SONDERHEFT 2/1986 ÖS 100,-/Sfr. 14,- Lit. 12000/hfl. 18,-/dkr. 68,- DM 14,-

HAPPY-COMPUTER

1. ATARI SONDERHEFT

HAPPY-COMPUTER

Markt&Technik

DAS GROSSE HEIMCOMPUTER-MAGAZIN



Alles für 800 XL und 130 XE:

Tips, Tricks und tolle Listings

Zum Abtippen: Das schnellste Basic

Interpreter und Compiler

Grundlagen

- ★ Musik
- ★ Compiler
- ★ Einsprungadressen

Basteleien

Hardware- und Software-Tests

lle Programme auch auf
iskette erhältlich

HAPPY-COMPUTER

PROGRAMM-SERVICE

Das Angebot dieser Sonderausgabe:

Wie auch zu unserer Stammzeitschrift Happy-Computer bieten wir wieder alle in die ·em Sonderheft veröffentlichten Programme auf Diskette an. Da wir aber nic· lle Programme auf einer Diskette unterbringen konnten, haben wir sie u. ·i Disketten im normalen DOS-2.0-Format verteilt. Somit kann man also ·uc ·as alte 810-Diskettenlaufwerk verwenden. Übrigens ist Diskette 1 beid-sei· Diskette 2 einseitig bespielt. Da sich auf allen drei Diskettenseiten j· ·ein DOS.SYS-File befindet, können Sie die Disketten booten. Dazu müs-s· nur die entsprechende Diskette ins Laufwerk schieben und die Strom-·gung des Computers einschalten.
·ie irgendwelche Programme ausprobieren, sollten Sie sich Sicherheits-·n anfertigen. Am einfachsten und schnellsten geschieht dies mit der ·ktion »J« vom DOS-Menü aus.
·enn Sie die Diskette 1 (mit dem Aufkleber nach oben) booten, gelangen Sie ·automatisch ins Turbo-Basic XL. Mit DIR erfolgt dann die Ausgabe des Inhaltsverzeichnisses auf dem Bildschirm. Sollten Sie noch den älteren Atari 800 verwenden, booten Sie bitte die zweite Seite von Diskette 1.
Im Atari-Sonderheft finden Sie Programme in normalem Atari-Basic, in Turbo-Basic XL, in Maschinensprache und in Assembler. Damit Sie die Programme auf Diskette besser zuordnen können, haben wir für die oben genannten Kriterien jeweils einen bestimmten Extender gewählt. Sie lauten im einzelnen:

XXXXXXXX.BAS Atari-Basic. Um solche Programme laden zu können, müssen Sie sich entweder in Atari-Basic oder in Turbo-Basic XL befinden. Achtung! Manche Basic-Programme können unter Turbo-Basic XL zu schnell ablaufen.

XXXXXXXX.TUR Turbo-Basic XL. Solche Programme müssen unbedingt unter Turbo-Basic XL laufen. Da dieses Basic über einen erweiterten Befehlssatz verfügt, ist es nicht kompatibel zum Atari-Basic. Sollten Sie dennoch ein solches Programm mit dem Standard-Atari-Basic laufen lassen, erhalten Sie entsprechende Fehlermeldungen.

XXXXXXXX.M65 MAC/65-Quelltexte. Solche Dateien müssen unbedingt mit dem MAC/65-Assembler assembliert werden. Mit Basic lassen sie sich nicht verarbeiten.

XXXXXXXX.COM Objekt-Code. Hierbei handelt es sich um Programme, die man sonst mit AMPEL (Atari-Maschinen-Programm-Eingabe-Listing) eingeben muß. Um solche Programme zu starten, wählen Sie im DOS-Menü die Funktion »L« und geben dann den entsprechenden Programmnamen ein. Anschließend wird das Programm geladen und automatisch gestartet. Wenn Sie sich in Turbo-Basic XL befinden, lassen sich solche Programme auch mit »BRUN"D:XXXXXXXX.COM« aufrufen und anschließend starten.

XXXXXXXX.SYS System-Dateien (DOS 2.5). Auf der Rückseite von Diskette 1 befindet sich außerdem noch ein DOS.SYS-File mit 145 Sektoren Länge. Hierbei handelt es sich um die abgespeckte Turbo-Basic XL-Version für den Atari 400 oder 800 mit mindestens 48 KByte-RAM. Wenn Sie diese Diskette booten, gelangen Sie automatisch ins Turbo-Basic.
Damit wären also die einzelnen Programmtypen erklärt. Sollten Sie Probleme mit den Programmen auf den Leserservice-Disketten haben, empfiehlt es sich, die entsprechenden Erläuterungen im Sonderheft genau und aufmerksam durchzulesen. So kann manch ein Problem gelöst werden.

Bestell.-Nr. LH 86S2D, DM 34,90*, (2 Disketten)
*inkl. MwSt. Unverbindliche Preisempfehlung.

Programme aus früheren Ausgaben

ATARI 800XL/130XE/800

Turbo-Basic
Der schnelle Basic-Interpreter für den Atari. Auf der Diskette befindet sich je eine Version für den Atari 800XL und eine für den Atari 800 mit mindestens 48 KByte-RAM. Aus Ausgabe 12/85.

AMPEL
Atari-Maschinen-Programm-Eingabe-Listing. Aus Ausgabe 12/85.

Atari-Prüfsummer
Eingabehilfe für alle in Happy-Computer veröffentlichten Basic-Programme.

Jumper II
Listing des Monats aus Ausgabe 8/84, um die Geschwindigkeit von Turbo-Basic zu demonstrieren.

Magic-Painter
Listing des Monats aus Ausgabe 3/85. Ein Zeichenprogramm, das an Turbo-Basic angepaßt wurde.

Alle 5 Programme auf einer Diskette für den ATARI 800XL/130XE/800.
Bestell-Nr. LH 8512B, DM 29,90*

ATARI 800XL

Prüfsummer
Eingabehilfe für alle in Happy-Computer veröffentlichten Basic-Programme.

Ge· ·lheimer
M· ·creen-Editor und 20 fertigen Szene· ·Spiel), aus Ausgabe 5/85.

24 ·ben in Grafikstufe 0
Rc· ·e für farbige Schrift (Utility), aus Au· ·be 6/85.

Di· ·elp
Für die schnelle Rettung (Utility), aus Ausgabe 8/85.

Ölsuche
Mit dem Atari auf Ölsuche (Spiel), aus Ausgabe 8/85.

Autostart
Basic-Programme automatisch starten (Utility), aus Ausgabe 9/85.

Dudu 4.0
Mehr Speicher mit der 1050 Floppy (Utility), aus Ausgabe 10/85.

Alle 7 Programme auf einer Diskette für den Atari 800 XL.
Bestell-Nr. LH 8510 B, DM 29,90*

ATARI 48 KByte-RAM

Magic Painter
Unser Listing des Monats in der Ausgabe 3/85 ist ein Grafikprogramm, das sich mit anderen Malprogrammen dieser Art durchaus messen kann. Besonders gelungen ist die einfache Bedienung, da man mit dem Joystick sowohl im Haupt- als auch in den Untermenüs sämtliche Punkte anwählen kann. Der elektronische Malkasten verfügt über 16 Menüpunkte und bietet eine Grafikauflösung von 160x96 Pixels.

Grafikdemo
Alle 256 Farben werden auf dem Bildschirm dargestellt. Eine Farbspielerei, die die hervorragenden Grafikfähigkeiten der Atari-Computer beweist (Rainbow-Effekt). Aus Ausgabe 3/85.

Variablen-Dump
Mit diesem Programm können Sie die verwendeten Variablen eines anderen Programms auf dem Bildschirm listen. Ein wichtiges Utility, das Ihnen die lästige Fehlersuche in längeren Basic-Programmen erleichtert. Aus Ausgabe 2/85.

Wie die Bilder laufen lernen
Mit dem Utility »Power-Mover« können Sie laufende Bilder schnell und problemlos erzeugen. Für alle, die sich an die Programmierung von Player-Missile-Grafiken heranwagen. Aus Ausgabe 2/85.

Statuszeile mit Uhr
Damit Sie beim Programmieren nicht die Zeit vergessen, hilft nur eine ständig sichtbare Zeitanzeige. Mit diesem Programm können Sie eine zusätzliche Statuszeile oberhalb des Bildschirms generieren. Aus Ausgabe 1/85.

Alle 5 Programme auf Diskette für Atari mit mindestens 48 KByte-RAM.
Bestell-Nr. LH 8503 B, DM 29,90*

ATARI 48 KByte-RAM

Diamantenfieber
Unser Listing des Monats aus der Ausgabe 2/85 ist eine wahre Schatztruhe. Bereichern Sie sich an bunt glitzernden Diamanten, die kunterbunt in einem Bergwerk verteilt sind. Aber Vorsicht! Die Stollen sind sehr unstabil. Eine falsche Bewegung genügt, und Sie werden von losen Gesteinsbrocken erschlagen. Ein Spiele-Designer sorgt bei unserer exzellenten »Boulder Dash«-Variante für anhaltende Spannung. Entwerfen Sie Ihre eigenen Bergwerkstollen, aber mit Bedacht, denn nicht selten ergibt sich bei einem Bild nur ein Lösungsweg.

Die Schatzhöhle
Wer möchte sich nicht auch mit einem Schatz bereichern. Wer dazu nicht unbedingt eine Weltreise unternehmen möchte, kann mit seinem Atari 800XL in eine Schatzhöhle eindringen. Gefährliche Tiere wie Skorpione, Ratten und Schlangen erschweren die Suche. Nur mit viel Geschick können Sie Ihren Geldbeutel – aber nur im Spiel – auffüllen. Aus Ausgabe 1/85.

Zeilenzauber
Die wichtigste RENUMBER-Funktion fehlt leider im Standard-Atari-Basic. Dieses Programm behebt diesen Mangel. Es ist leicht zu bedienen. Einfach das Basic-Programm von der Diskette laden, »RUN« eingeben, und das entsprechende Objekt-File wird auf Diskette geschrieben. Dann nur noch vom DOS-Menü aus mit der Funktion »L« das Objekt-File laden und mit »B« zurück ins Basic gehen. Wenn Sie ein Basic-Programm umnumerieren möchten, geben Sie einfach »PRINT USR (8044)« ein, und Ihr Programm wird wunschgemäß umnumeriert. Aus Ausgabe 11/84.

Jumper II
Ein professionell gemachtes, in Basic geschriebenes Spiel. Exzellente Programmierung, ein Highscore-Zähler und eingebautes Demo werden selbst Zweifler schnell überzeugen. Auf musikalische Untermalung wurde großer Wert gelegt, und die verschiedenen Screens sind brillant gemacht. Genau das richtige Programm für kalte Winterabende. Listing des Monats. Aus Ausgabe 8/84.

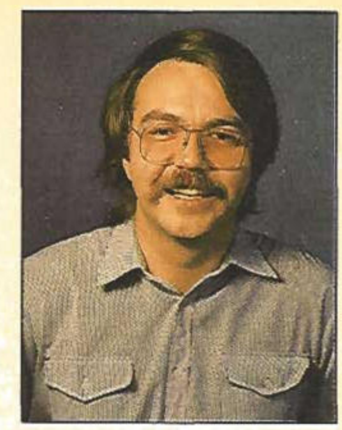
Mop – Der Goldgräber
Schnelligkeit und guter Sound zeichnen dieses Spiel aus. Die Soundfähigkeiten sind wirklich hervorragend. Viele Bilder sorgen bei diesem Programm für viel Abwechslung. Ausgabe 7/84.

Alle 5 Programme auf Diskette für Atari mit mindestens 48 KByte-RAM.
Bestell-Nr. LH 8502 B, DM 29,90*

Bestellungen aus der Schweiz richten Sie bitte direkt an: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042/415656.
Bestellungen aus anderen Ländern bitte per Auslandspostanweisung! Achtung: Nicht die eingeheftete Zahlkarte verwenden!
Bestellungen aus Österreich richten Sie bitte direkt an: Ueberreuter Media Handels- und Verlagsges. mbH, Alser Str. 24, 1091 Wien, Tel. 0222/481538-0

* Alle Preise inklusive Mehrwertsteuer, unverbindliche Preisempfehlung, Leser-Service-Produkte sind nur für Endkunde·., nicht für Wiederverkäufer.





Atari im Aufwind

Als Atari-Besitzer kommt man sich manchmal wie auf einer einsamen Südsee-Insel vor. Sucht man Informationen oder Listings für seinen 800XL oder 130XE in deutschen Computerzeitschriften, sitzt man leider allzuoft auf dem Trockenen. Diesem Notstand wollen wir mit diesem Sonderheft abhelfen. Hier finden Sie eine Fülle von Listings und Informationen, die Sie hoffentlich für die lange Durststrecke entschädigen. Der Atari 800XL/130XE zählt schließlich nach wie vor zu den leistungsfähigsten derzeit erhältlichen Heimcomputern. Bis zu 256 gleichzeitig darstellbare Farbschattierungen und die Player Missile-Grafik, mit der sich einige Effekte viel besser umsetzen lassen als mit Sprites, sind nicht zu übersehen. Berücksichtigt man dann den sehr günstigen Anschaffungspreis des Atari 800XL/130XE, ist dessen Preis/Leistungsverhältnis wirklich von kaum einem Heimcomputer zu schlagen. Auch die Peripherie, also Disketten-Laufwerke und Drucker, ist in letzter Zeit preiswert geworden.

Da wären aber noch die neuen 16-Bit-Computer. Man weiß, daß der Atari 800XL und der 130XE mit einer 8-Bit-CPU vom Typ 6502 ausgestattet ist. Damit scheint klar zu sein, daß ein 16-Bit-Prozessor viel leistungsfähiger sein muß als das 8-Bit-Gegenstück. Zählt also der 800XL im Vergleich zum ST zum alten Eisen? Sicher kann der 800XL nicht mit der Grafikauflösung oder der Geschwindigkeit des ST konkurrieren. Aber einen ganz großen Pluspunkt gibt es für den 800XL: nämlich Software für jeden Zweck. Als Spielecomputer zum Beispiel, ist der ST zur Zeit noch keine Alternative. Natürlich wird es in Zukunft viel Software für diesen Computer geben, aber wer sofort loslegen möchte und vorzugsweise auf fertige Programme Wert legt, ist mit dem 800XL besser bedient.

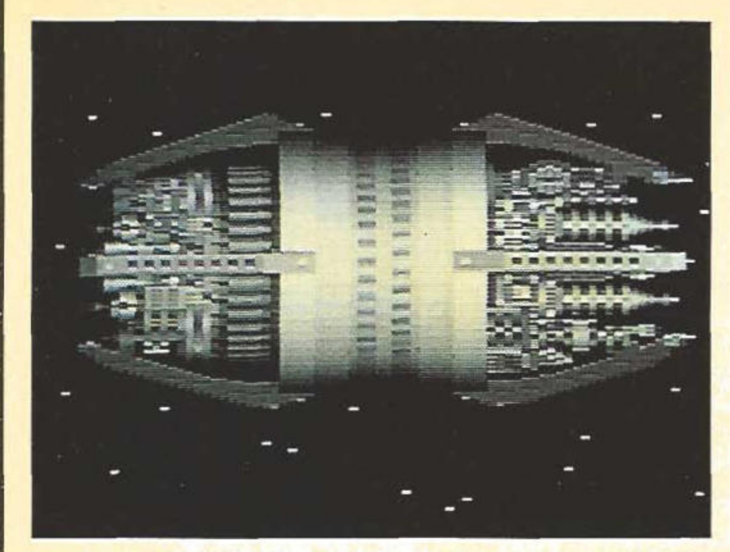
Als wirklich hervorragendes Beispiel für ausgezeichnete Software mag unser Turbo-Basic-Interpreter mit zugehörigem Compiler dienen. Beide Programme finden Sie in diesem Sonderheft – natürlich zum Abtippen. Sie brauchen also nur ein wenig Zeit zu investieren und haben anschließend einen Interpreter und einen Compiler, die es in sich haben. Geschwindigkeitsvergleiche haben nämlich ergeben, daß Turbo-Basic XL seinem Namen alle Ehre macht; dieser Interpreter zählt zu den schnellsten, die es für den Atari gibt. Der dazu passende Compiler compiliert sogar in normalem Basic geschriebene Programme. Diese Programme sind dann wirklich die absoluten »Renner«. Und damit Sie Ihr Turbo-Basic XL auch gleich ausprobieren können, haben wir einige Programme in dieser Programmiersprache veröffentlicht.

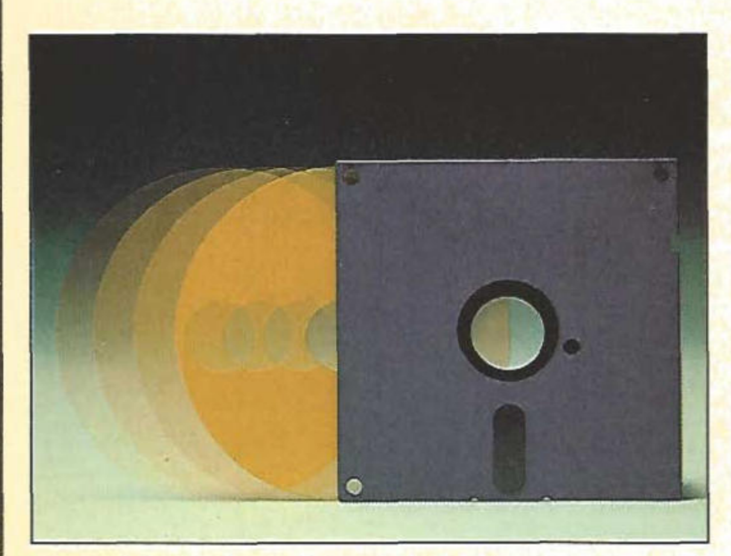
Warum sollte man also neidisch werden, wenn in den meisten Zeitschriften mehr über andere Computer berichtet wird. Schließlich wird man einige Zeit brauchen, bis man alle in diesem Sonderheft veröffentlichten Programme ausprobiert hat. Oder bauen Sie sich doch die eine oder andere Hardwarebastelei nach. Sei es eine Zehnertastatur oder ein Cartridge-Experimentier-System, das Angebot ist reichhaltig und sorgt sicher für einige unterhaltsame Stunden.

Ebenfalls in diesem Sonderheft finden Sie eine Umfrage, bei der es eine Menge schöner Preise als Belohnung fürs Mitmachen zu gewinnen gibt. Wir möchten nämlich gerne Ihre Meinung zu diesem Sonderheft wissen. Schließlich brauchen wir Ihre Anregungen und Vorschläge, um weitere Sonderhefte und natürlich vor allem unsere Stammzeitschrift Happy-Computer für Sie so interessant wie möglich zu gestalten. Unter allen Einsendungen verlosen wir unter anderem zwei Atari 1050 Disketten-Laufwerke und einen Atari 1029 Nadel-Matrixdrucker. Also nicht lange zögern! Fragebogen ausfüllen, in ein Kuvert stecken und ab die Post an die Redaktion. Die Gewinner werden dann in einer der nächsten Ausgaben von Happy-Computer bekanntgegeben.

Werner Breuer

INHALT





Diskettenservice

Wer keine Zeit oder keine Lust hat, alle Programme selbst in mühevoller Kleinarbeit abzutippen, kann wieder auf den bewährten Diskettenservice zurückgreifen. Es sind hier sämtliche Programme des Sonderhefts auf zwei Disketten erhältlich.

Bestellnummer LH 86 S2D **29,90 DM** (2 Disketten)

Hardware-Tests

Basteleien

Das schnellste Basic

Zum Abtippen:

Anwendungs-Listing

Spiele-Listings

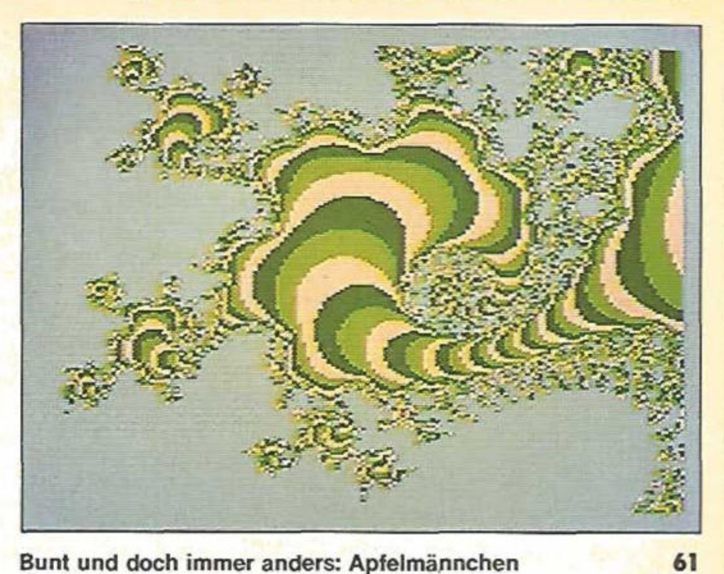
Eingabehilfen

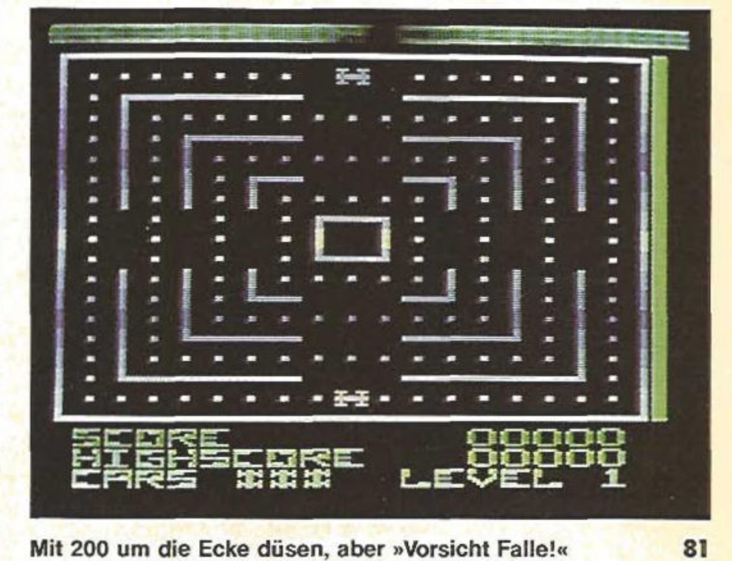
Tips & Tricks

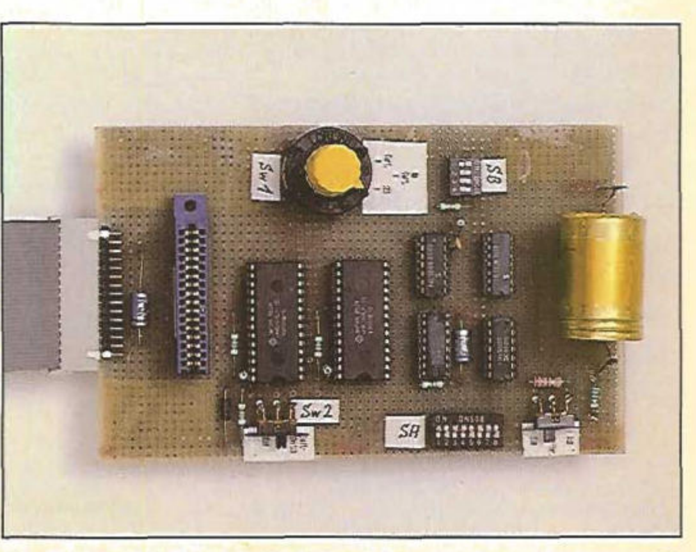
Grundlagen

Software-Tests

Rubriken







Auf die Verbindung kommt es an

Schnittstellen stellen die Verbindung zwischen Computer und Außenwelt dar. Wir haben zwei Interfaces unter die Lupe genommen.

Will man seine Programme nicht nur auf dem Bildschirm bewundern, so benötigt man einen Drucker und daneben auch eine passende Schnittstelle. Beim Atari kann man dieses Problem auf zwei Arten lösen. Besitzt man einen Original-Atari-Drucker, wie zum Beispiel den Atari 1029, so ist keine separate Schnittstelle nötig. Diese Drucker sind nämlich dafür ausgelegt, ihre Daten über den seriellen Bus des Atari zu empfangen. Der Nachteil dabei ist, daß solche Drucker wirklich nur für den Atari verwendungsfähig sind. Bei einem Systemwechsel muß man sich dann zwangsläufig einen neuen Drucker anschaffen.

Die Alternative hierzu ist ein beliebiger Drucker mit Centronics-Schnittstelle. Diese Schnittstelle entspricht einer bestimmten Norm und hat die Eigenschaft, Daten parallel zu verarbeiten. Die Datenübertragung des Atari jedoch erfolgt seriell. Daher muß eine Schnittstelle die Wandlung des Datenformats vornehmen. Bis vor etwa zwei Jahren gab es deshalb bei Atari auch das Interface 850 zu kaufen. Neben einer Centronics-Schnittstelle waren hier auch vier serielle Schnittstellen vom Typ RS232 eingebaut. Daran lassen sich vor allem Akustikkoppler und auch einige Drucker anschließen. Dieses Interface ist jedoch nicht mehr erhältlich, und so übernahmen es verschiedene Fremdhersteller, die Schnittstellen für den Atari anzubieten.

Eine dieser Schnittstellen nennt sich Interface 850 XL und kostet rund 400 Mark inklusive Netzteil und Druckerkabel. Neben einer Centronics-Schnittstelle ist zwar nur eine RS232-Schnittstelle eingebaut, diese genügen allerdings für die meisten Anwendungen. Der RS232-Teil des Interfaces ist voll kompatibel zur alten Atari-850-Schnittstelle. Den Centronics-Teil jedoch hat man etwas erweitert. Ein Nachteil der 850-Schnittstelle gegenüber der Verwendung der Original-Atari-Drucker war nämlich die fehlende Codewandlung der Grafikzeichen. Nicht jeder Drucker, der an die Centronics-Schnittstelle angeschlossen werden kann, ist auch in der Lage, Grafikzeichen zu drucken (zum Beispiel Typenraddrucker). Darum werden alle Codes unverändert an den jeweiligen Drucker geschickt. Dieser Nachteil läßt sich nur durch entsprechende Treibersoftware wieder beheben.

Beim Interface 850 XL hat man nun versucht, diesen Nachteil direkt mit der Schnittstelle wieder wettzumachen. Das Interface ist in vier verschiedenen Druckmodi zu betreiben, die über eine kurze Steuersequenz ausgewählt werden. Im ersten Modus werden alle Zeichen ohne Codewandlung an den Drukker geschickt. Im zweiten Modus, der standardmäßig eingestellt ist und dem 850-Interface-Modus entspricht, wird nur der »Carriage Return«-Code umgewandelt. Der Drucker benötigt zum Zeilenvorschub nämlich den Code 13, während der Atari dazu den Code 155 verwendet. Der dritte Druckmodus ist für Drucker gedacht, die nicht grafikfähig sind. Alle Codes, die beim Atari mit Grafikzeichen belegt sind, werden einfach in Standard-ASCII-Codes umgewandelt. Druckt man nun beispielsweise ein Listing aus, in dem Grafikzeichen vorkommen, so erscheinen diese zwar nicht in ihrer richtigen Form, der Drucker jedoch interpretiert diese Zeichen wenigstens nicht als Steuercodes. Wie der Zeichensatz des Atari in diesem Modus aussieht, sehen Sie in Bild 1. Der vierte Modus ist für grafikfähige Drucker gedacht. Mit jedem Grafikzeichen wird auch ein entsprechender Steuercode an den Drucker geschickt, der ihn in den Grafikmodus umschaltet. Leider war man hierbei nicht ganz konsequent. Alle invertierten Zeichen werden nicht invertiert gedruckt (Bild 2). Bei normalen Basic-Listings fällt dies nicht weiter ins Gewicht. Verwendet man aber beispielsweise in einem Programm Maschinenunterroutinen, die als Strings und somit in Form von Grafikzeichen abgelegt sind, enthält das Druckerlisting schlichtweg falsche Informationen.

Neben den beiden Schnittstellen enthält das Interface 850 XL auch noch ein Kassettenrecorder-Interface. Man kann so statt des Atari-Datenrecorders auch jeden anderen handelsüblichen Kassettenrecorder an den Atari anschließen.

Benötigt man keine RS232-Schnittstelle, kann man sich auch ein reines Druckerinterface, also nur mit Centronics-Schnittstelle, kaufen. Zum Beispiel das Atari-Interface 72000 zum Preis von rund 250 Mark. Es benötigt kein Netzteil, da es direkt vom Atari mit Strom versorgt wird. Das Gehäuse ist im Druckerkabel integriert. Mit Hilfe von Dipschaltern kann man auch hier zwei verschiedene Druckmodi einstellen. Der erste Modus schickt, mit Ausnahme des »Carriage Return«-Zeichens, alle Codes unverändert weiter und entspricht somit wieder dem 850-Interface von Atari. Der zweite Modus ist speziell für Besitzer des Atari-Schreibers interessant. Dieses Textverarbeitungsprogramm benutzt recht seltsame Codes für die deutschen Umlaute. Bei manchen Druckern kann es daher zu Schwierigkeiten beim Ausdruck kommen. Durch eine entsprechende Codewandlung hebt dieses Interface diesen Nachteil auf.

Neben diesen beiden Interfaces sind vor allem aus den USA noch eine Reihe von Druckerschnittstellen, die mehr oder weniger dieselben Funktionen ausüben, auf dem Markt. Generell kann man sagen, daß die Anschaffung eines Druckerinterfaces und eines normalen Druckers dem Kauf eines Atari-Druckers vorzuziehen ist. Dies gilt vor allem dann, wenn man plant, sich im Laufe der Zeit einen anderen Computer anzuschaffen.

(Wolfgang Czerny/wb)

Bezugsquellen:

Kunkel Industriebedarf, Zweibrückener Str. 8, 7000 Stuttgart 31, Tel. (07 11) 8847 11

Reinhard Wiesemann, Winckenbachstr. 3a, 5600 Wuppertal 2, Tel. (0202) 505077

```
§ABCDEFGHIJKLMNOPQRSTUVWXYZÄöÜ^_ !"#$%&'()*+,-./0123456789:;<=>?§ABCDEFGHIJKLMNO
PQRSTUVWXYZÄöÜ^_`abcdefghijklmnopqrstuvwxyzäöü§ABCDEFGHIJKLMNOPQRSTUVWXYZ
öÜ^_ !"#$%&'()*+,-./0123456789:;<=>?§ABCDEFGHIJKLMNOPQRSTUVWXYZÄöÜ^_`abcdefghijk
lmnopqrstuvwxyzäöü
```

Bild 1. Beispielsausdruck mit dem 850XL-Interface. In Modus 3 werden sämtliche Steuer- und Grafikzeichen in ASCII-Zeichen umgewandelt.

```
♥├ ┝┤┐╱╲◢▃▲▀▔▁▖♣┌─┼●▄ ┬┴▌ └←↑↓↔ !"#$%&'()*+,-./0123456789:;<=>?§ABCDEFGHI
JKLMNOPQRSTUVWXYZÄöÜ^_♦abcdefghijklmnopqrstuvwxyz♠|↖◀▶♥├ ┝┤┐╱╲◢▃▲▀▔▁▖♣┌─┼●▄
|┬┴▌ └
↑↓↔ !"#$%&'()*+,-./0123456789:;<=>?§ABCDEFGHIJKLMNOPQRSTUVWXYZÄöÜ^_♦abcdefghij
klmnopqrstuvwxyz♠|↖◀▶
```

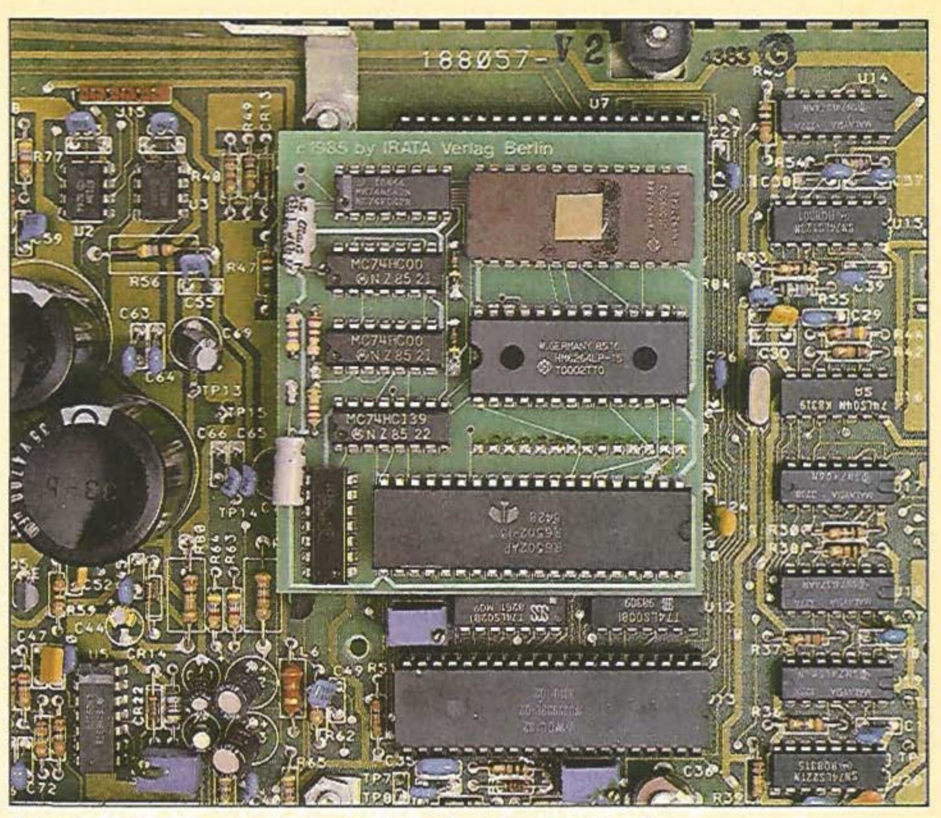
Bild 2. In Modus 4 lassen sich auf einem grafikfähigen Drucker auch Steuer- und Grafikzeichen wiedergeben.



Rasende Daten

Die Erweiterung »High Speed 1050« macht ihrem Laufwerk Beine und erhöht auch noch die Speicherkapazität.

Der Floppy-Speeder »High Speed 1050« für das Atari 1050-Laufwerk besteht lediglich aus einer kleinen Erweiterungsplatine. Auf dieser Platine befinden sich neben einigen Gatterbausteinen auch ein neuer EPROM, ein RAM-Baustein und ein Prozessor vom Typ 6502. Zum Einbau der Erweiterung muß man als erstes das Gehäuse der Diskettenstation öffnen. Dann hebt man vorsichtig das Laufwerk aus seiner Halterung. Nun liegt die Hauptplatine, auf der sich der alte ROM- und der alte RAM-Baustein befinden, offen da. Diese beiden ICs werden aus ihren Sockeln gezogen und statt dessen die Erweiterungsplatine in eine der frei gewordenen Fassungen gesteckt. Der Einbau ist hiermit bereits beendet. Es versteht sich von selbst, daß man dabei größte Vorsicht walten lassen muß. Der Garantieanspruch erlischt nämlich, sobald die Abschirmung der Hauptplatine entfernt wurde.



Acht Bauteile verwandeln ein 1050-Laufwerk in eine Rennfloppy

Schnell wie der Wind

Ein auf diese Weise umgebautes Laufwerk hat nun mehrere Vorteile. Zunächst sorgt das neue EPROM dafür, daß man Disketten nun auch mit doppelter Dichte formatieren kann. Das normale 1050er-Laufwerk kann ja neben der einfachen Schreibdichte bereits mit erhöhter Schreibdichte formatieren. Bei der einfachen Dichte wird mit 720 Sektoren von je 125 Byte gearbeitet. Das ergibt eine Speicherkapazität von 88 KByte. Bei der erhöhten Schreibdichte steigt die Zahl der Sektoren auf 1040 an, pro Sektor sind aber immer noch 125 Byte vorgesehen. Die Speicherkapazität ist also auf 126 KByte gestiegen. Doppelte Dichte bedeutet nun, daß man zwar wieder nur über 720 Sektoren, diesmal jedoch von je 256 Byte und somit über 180 KByte Speicherplatz verfügt.

Die Formatierung auf doppelte Schreibdichte erfordert natürlich auch ein neues DOS. Aus diesem Grund bekommt man zusammen mit der Erweiterungsplatine auch gleich eine Diskette mit zwei verschiedenen DOS-Versionen mitgeliefert. Bei »Warp-Speed-DOS« handelt es sich um ein erweitertes DOS 2.0, das wahlweise Disketten mit einfacher oder doppelter Dichte verarbeitet. Die andere DOS-Version mit dem Namen »MYDOS« ist auch interessant. Hiermit kann man auch wieder unterschiedliche Schreibdichten formatieren. Daneben lassen sich beispielsweise aber auch Subdirectories anlegen.

Alle Daten, die gespeichert oder geladen werden, werden in dem neu hinzugekommenen RAM-Baustein zwischengespeichert. Ein Test ergab, daß sich bis zu 1875 Byte in diesem RAM unterbringen lassen. Speziell bei Dateiverwaltungsprogrammen kann es daher vorkommen, daß Datensätze von Diskette gelesen werden, ohne daß das Laufwerk tatsächlich auf die Diskette zugreifen muß. Der Schreib-/Lesekopf wird also bei Programmen, die häufigen Diskettenzugriff erfordern, geschont. Andererseits hat diese Sache auch einen Haken. Bearbeitet man eine Datei und schließt man den einen Schreibvorgang nicht korrekt mit CLOSE ab, so bleiben die Diskettenanweisungen im Puffer. Wechselt man nun die Diskette im Laufwerk, kann es passieren, daß die im Puffer verbliebenen Anweisungen auf die neue Diskette geschrieben werden und dort Files zerstören.

Durch diese Erweiterung wird das Laufwerk auch um einiges schneller. Vor allem beim Lesen von Daten sind Geschwindigkeitssteigerungen um 50 Prozent keine Seltenheit. Um welchen Faktor sich die Geschwindigkeit im einzelnen erhöht, ist dabei stark vom jeweiligen Lesebefehl abhängig. So verringert sich die Zeit, die das Laufwerk beispielsweise zum Booten des DOS.SYS-Files benötigt, nur um rund eine Sekunde. Liest man aber zum Beispiel 1000 Zahlen mit dem INPUT-Befehl, so sind dazu statt rund 35 Sekunden nur mehr rund 20 Sekunden nötig. Das Formatieren einer Diskette beansprucht nur noch 26 Sekunden statt 38 Sekunden. Speichert man Programme mit dem LIST-Befehl auf Diskette, so hält das Laufwerk sogar zwischenzeitlich immer wieder an, da die Datenübertragung langsamer ist als das Speichern.

Formatieren in 26 Sekunden

Diese Geschwindigkeitssteigerung hat allerdings auch einen Nachteil. Manche professionellen Programme lassen sich nicht mehr laden, da ihr Kopierschutz auf die veränderte Zugriffszeit reagiert. Mit geeigneter Software wäre es allerdings möglich, das Laufwerk wieder zu normalisieren. Dann stellt das Laden dieser Programme mit dem erweiterten Laufwerk auch kein Problem mehr dar.

Die rund 230 Mark, die man für die Erweiterung ausgeben muß, sparen also nicht nur die Hälfte an Disketten ein, sondern bieten zusätzlich ein schnelleres Laufwerk. Somit ist der Einbau durchaus zu empfehlen.

(Wolfgang Czerny/wb)

Bezugsquelle: IRATA, Hermannstr. 9, 1000 Berlin 44, Tel.: (030) 6212071

Die Raupe wird zum Schmetterling

Um in den Genuß von 64 KByte RAM mehr Speicherplatz zu kommen, muß man sich nicht unbedingt einen Atari 130XE kaufen. Eine Platine verwandelt Ihren 800XL in einen 130XE.

Für den Atari 800XL gibt es jetzt eine Erweiterungsplatine, durch die der 800XL voll kompatibel zum 130XE wird. Um es allerdings gleich vorweg zu sagen: Wer Veränderungen an der Hardware seines Atari 800XL vornimmt, riskiert natürlich den Verlust des Garantieanspruchs. Man sollte den Einbau dieser Platine also entweder dem Fachmann überlassen oder aber entsprechende Grundkenntnisse mitbringen.

Auf der neuen Platine befinden sich neben einigen Gatterbausteinen sechs zusätzliche RAM-Bausteine. Die Erweiterung hat ihren Platz direkt auf der Hauptplatine des Atari 800XL. Das Gehäuse muß also geöffnet und das Abschirmblech entfernt werden. Die alten RAM-Bausteine werden aus ihren Sockeln gezogen und in die dafür vorgesehenen Sockel auf der Erweiterungsplatine eingesetzt. Die Platine selbst wird in zwei der so freigewordenen Sockel auf der Hauptplatine gesteckt. Nun sind noch einige Drähte von der Erweiterungsplatine an verschiedenen Stellen auf der Hauptplatine anzuschließen. Da Lötarbeiten vermieden werden sollten, schließt man die Drähte direkt an verschiedene ICs an. Dies geschieht, indem man die Drahtenden zusammen mit den IC-Anschlüssen in die Sockel klemmt. Manche Fälle erfordern aber auch ein Hochbiegen der Anschlüsse der ICs, so daß man die Drähte direkt anklemmen kann.

Beim Einbau ist etwas Fingerspitzengefühl unumgänglich. Sonst kann es allzuleicht passieren, daß man versehentlich ein IC-Beinchen abbricht oder das eine oder andere IC durch statische Aufladung zerstört. Aus diesem Grund sollte man vielleicht überlegen, ob man die Erweiterungsplatine nicht doch vom Hersteller der Platine in seinen Atari einbauen läßt.

Ist die Platine an ihrem Platz, verfügt man über einen zusätzlichen RAM-Speicherbereich von 64 KByte. Leider sind diese 64 KByte nicht genauso ansprechbar wie der normale RAM-Speicher. Da der Atari als 8-Bit-Computer nur 64 KByte direkt adressieren kann, muß man den Zusatzspeicher über das sogenannte »Bank-Switching«-Verfahren ansprechen. Gibt man in Basic also beispielsweise den FRE(0)-Befehl ein, hat sich am freien Speicherplatz scheinbar nichts geändert.

Der Zusatzspeicher ist in 4 Banken aufgeteilt, die zu erreichen es einen kleinen Umweg erfordert. Das Bild zeigt, wie die Speicherbereichsverteilung unter 128-KByte-RAM aussieht. Jede der vier Banken umfaßt 16 KByte und muß separat adressiert werden. Diese Aufgabe übernimmt beim 800XL der PIA-Baustein. Dieser Interface-Baustein verfügt über zwei 8 Bit breite Ports. Bei den alten Atari 400 und 800 übernahmen diese beiden Ports die Abfrage der Joysticks. Beim 800XL hingegen wurden zwei Joystickports eingespart. Der dadurch freigewordene zweite Port der PIA wird nun zur Steuerung der MMU (Memory Management Unit oder Speicherverwaltungs-Einheit) eingesetzt. POKEt man also einen entsprechenden Wert in das Portregister, hat man der MMU so mitgeteilt, welche Bank angesprochen werden soll.

Speicherriese 800XL

Der Zusatzspeicher eignet sich also weniger zur Aufnahme größerer Basic-Programme, sondern vielmehr dazu, um darin Daten abzulegen; Daten wie zum Beispiel Maschinenunterprogramme oder Player/Missile-Daten. Man kann natürlich auch mehrere Grafik-Bilder darin ablegen oder ganz einfach allgemeine Datenfelder. Der Vorteil liegt klar auf der Hand, schreibt man ein umfangreiches Programm, kann manches Nachladen von Diskette entfallen. Programme wirken dann nicht mehr »langatmig« und der Anwender wird nicht mehr einer Geduldsprobe unterzogen. Arbeitet man beispielsweise mit dem Atari-DOS 2.5, ist es ohne weiteres möglich, in diesem Speicherbereich eine RAM-Disk anzulegen. Diese RAM-Disk kann dann wie ein Diskettenlaufwerk angesprochen werden. Man speichert oder lädt seine Programme nur mit der Gerätebezeichnung »D8:«. Da beim Laden und Speichern von Programmen in die RAM-Disk keine Input/Output-Operationen, sondern lediglich Speicherverschiebungen durchgeführt werden, gewährleistet dies so natürlich ein viel schnelleres Arbeiten. Befindet sich das DUP.SYS-File beispielsweise in der RAM-Disk, ist man beim Aufruf des DOS-Befehls innerhalb von Sekundenbruchteilen bereits im DOS-Menü.

Der Ausbau eines Atari 800XL auf 128 KByte RAM ist also eine durchaus lohnenswerte Angelegenheit. Lobenswert ist auch die Tatsache, daß der Selbsteinbau sowie der Einbau vom Hersteller das gleiche, nämlich 259 Mark kostet. Sollten Sie also kein erfahrener Bastler sein, ist es ratsam, den Ausbau vom Hersteller vornehmen zu lassen. Schließlich kostet es nicht mehr und eine einwandfreie Funktionsweise des Computers ist auch gewährleistet. Billiger als ein neuer 130XE ist dies auf jeden Fall.

(Wolfgang Czerny/wb)

Bezugsquelle: Wilhelm Bock, Bleichstr. 5, 4790 Paderborn, Tel. (05251) 32691

Adressen	Normaler RAM- und ROM-Bereich	Erweiterter RAM-Bereich
65536-$FFFF 42192-$C000	Antic, Pokey, Gtia, Betriebssystem ROMs	
42191-$BFFF 32768-$8000	Basic oder RAM Bildschirmspeicher	
32767-$7FFF	RAM	Bank 3 $C000-$FFFF
		Bank 2 $8000-$BFFF
		Bank 1 $4000-$7FFF
16384-$4000		Bank 0 $0000-$3FFF
16383-$3FFF 00000-$0000	Betriebssystemvariablen usw.	

Die Speicheraufteilung mit 128 KByte-RAM



ACHTUNG Atari 800XL- und Atari 130XE-Fans!

Das Listing des Monats im März-Heft von »Happy-Computer« bringt: »Happy-DOS II + /D mit RAM-Disk« kompatibel zu DOS 2.0.

Außerdem stellen wir Ihnen drei Monitore für den Betrieb am Atari-ST vor und berichten über das aktuelle Computergeschehen auf der CES in Las Vegas. Eine große Marktübersicht informiert Sie über das aktuelle Taschencomputerangebot samt zugehöriger Software. In einem Grundlagenbeitrag lernen Sie die logischen Operationen und deren erfolgreiche Anwendung in eigenen Programmen kennen.
Commodore-Fans und -Interessenten finden unter anderem einen Testbericht über das Super-Grafikprogramm »Deluxe Paint« für den Amiga. Wer sich für Schneider-Computer interessiert, sollte die große Marktübersicht »Hard- und Software für Schneider CPC 464, 664 und 6128« nicht versäumen.





Gutschein

FÜR EIN KOSTENLOSES PROBEEXEMPLAR VON HAPPY COMPUTER

JA, ich möchte »Happy-Computer« kennenlernen.
Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »Happy-Computer« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte »Happy-Computer« dann regelmäßig frei Haus per Post und bezahle pro Jahr nur DM 66,— statt DM 72,— Einzelverkaufspreis (Ausland auf Anfrage).

Vorname, Name

Straße — PLZ, Ort

Datum — 1. Unterschrift

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.

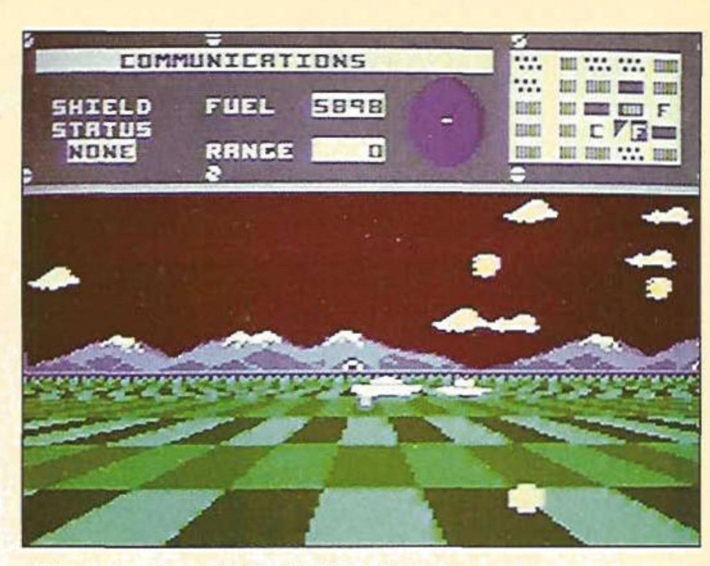
Datum — 2. Unterschrift

Gutschein ausfüllen, ausschneiden, in ein Kuvert stecken und absenden an: Markt&Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar

HCS286



Die Angst des Raumfahrers vor der Untertasse: »Koronis Rift«



»Dimension X« setzt Grafik-Maßstäbe

Spielmaschine Atari

Daß der Atari bis heute einen Ruf als Spielcomputer genießt, hat seine guten Gründe. Wenn auch Sie mit Ihrem Computer gerne eine Runde spielen möchten, finden Sie hier einige allgemeine Tips.

Spiele-Freaks, die einen 800XL oder 130XE besitzen, haben momentan allen Grund zu leichtem Frust. Die beiden Atari-Computer sind bekanntlich Sound- und Grafik-Computer allerhöchster Güte. Leider vernachlässigen die Softwarefirmen seit ein paar Jahren die Atari-Heimcomputer. Es erscheinen bei weitem nicht so viele Programme wie für den Marktführer Commodore 64. Hin und wieder gibt es auch Umsetzungen von C 64-Titeln, deren Qualität allerdings mitunter sehr zu wünschen übrigläßt, und die auch die Hardware nicht richtig ausreizen.

Frisch aus den USA im Direktimport

Der Atari-Spieler hat kein leichtes Leben, da in letzter Zeit relativ wenig Software-Nachschub anrollt. Bis auf U.S. Gold, die viele Atari-Versionen amerikanischer Programme im Angebot haben, vernachlässigen die englischen Softwarehäuser XL und XE geradezu sträflich. In den USA werden neue Spiele zwar bevorzugt für C 64 und Apple II entwickelt, doch gibt es eine ganze Reihe von Atari-Umsetzungen, die aber hierzulande nur sehr schwer zu erhalten sind. So kann man beim Compy-Shop zum exklusiven Preis von je 149 Mark einige direkt importierte Atari-Programme kaufen, die nur vorerst oder generell nicht in Deutschland vertrieben werden. Auf der jüngsten Angebotsliste findet man hier Leckerbissen wie den neuen U-Boot-Simulator von Microprose, »Silent Service«, die Synapse-Adventures »Mindwheel«, »Essex« und »Brimstone« sowie das Autorennen »Pitstop II«.

»Winter Games« im Anmarsch

Etwas billiger kommen Sportspiel-Fans weg. Zwei der drei erfolgreichen Olympia-Simulationen von Epyx gibt es auch auf Diskette für Atari-Computer: »Summer Games« und »Winter Games« kann man zum erschwinglichen Preis von knapp 60 Mark kaufen. Die Atari-Versionen enthalten die Original-Disziplinen, sind grafisch aber nicht ganz so stark.

Das Programmierteam von Lucasfilm Games hat erst vier Spiele veröffentlicht, doch ein Titel ist besser und erfolgreicher als der andere: »Rescue on Fractalus«, »Ballblazer«, »The Eidolon« und »Koronis Rift«. »Koronis Rift« ist die ausgereifteste und interessanteste Neuerscheinung. Als Raumkapitän düsen Sie mit Ihrer Lieblingsuntertasse durchs All. Plötzlich schlagen alle Instrumente aus: Sie haben Koronis Rift, den sagenumwobenen Planeten einer alten Rasse, wiederentdeckt. Im Raumgleiter rasen Sie über die Planetenoberfläche, um alte Raumschiff-Wracks nach wertvollen Überbleibseln der Supertechnik der verschollenen Ureinwohner zu durchsuchen. Mit diesen Teilen kann man dann das eigene Schiff besser ausrüsten, was auch nötig ist, wenn man den massiven Angriffen der Wächter-Flotten auf Dauer widerstehen will.

Neues von den Atari-Experten

Vor allem begeistert das Spiel mit seiner Grafik: Die Gebirgsketten von Koronis Rift ziehen perspektivisch perfekt an Ihrem Cockpit vorbei, und an Bord Ihres Mutterschiffs wacht ein toll animierter Roboter über die Beutestücke. Auch bei den anderen drei Lucasfilm-Spielen sind spannende Handlung und schnelle Grafik Trumpf. »Ballblazer« ist eine Art Fußballspiel des 4. Jahrtau-

sends, in dem die Bälle mit einem Magnetfeld eingefangen werden. »Rescue on Fractalus« ist ähnlich wie »Koronis Rift« eine Planeten-Expedition, bei der diesmal abgeschossene Raumpiloten vor den finsteren Jaggis gerettet werden müssen. Der Vorspann dieses Spiels gehört übrigens mit zum besten, was man je an animierter Grafik auf einem Heimcomputer gesehen hat.

Abenteuerspiele in Deutsch und Englisch

»The Eidolon« schließlich ist eine Expedition durch ein labyrinthartiges Höhlensystem, in dem es von merkwürdigen Kreaturen nur so wimmelt.

Abenteuerlustige Atari-Fans sitzen keinesfalls auf dem trockenen. Selbst wenn Sie keine Diskettenstation haben, können Sie mit Ihrem Computer ein saftiges Adventure spielen. Das englische Softwarehaus Level 9 ist auf Abenteuerspiele spezialisiert und bietet auch einige Programme für die Ataris an. Zirka 30 Mark kostet Sie der Spaß eines Grafik-Adventures mit guten, allerdings englischen Texten und vielen Bildern. Nur auf Floppy gibt es das hervorragende Einsteiger-Adventure »The Dallas Quest« (Top-Grafik und Top-Spielwitz, 59 Mark) und die Gehirnzellen-Feger »Mask of the Sun« und »Serpent's Star« (je 79 Mark). Der Bestseller »Mask of the Sun« liegt auch in einer voll eingedeutschten Version vor und nennt sich hier »Das Geheimnis der Aztekenmaske«.

Neben Tops auch Flops

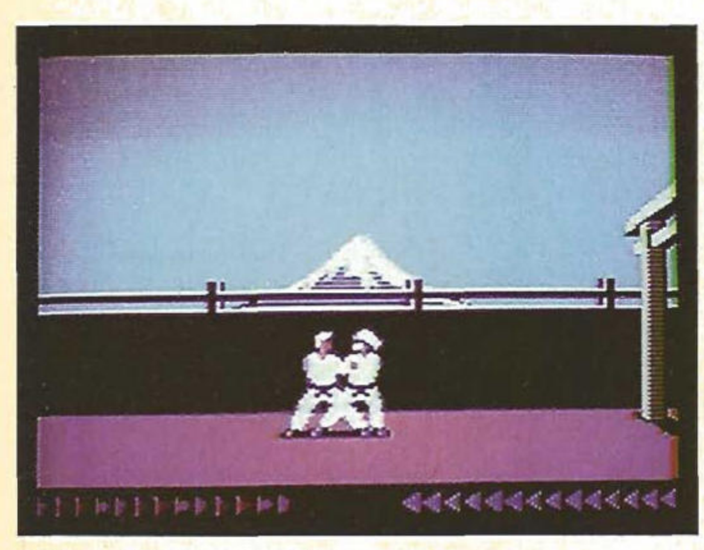
An dieser Stelle mal ein paar Warnungen vor Neuerscheinungen der letzten Monate, die viel Spielwitz versprechen, aber nur wenig halten. »Super Zaxxon«, die Fortsetzung zu »Zaxxon«, ist ein grafisch guter, aber spielerisch ausgesprochen liebloser Krawumm-Aufguß. Das Jump-and-Run-Spiel »Conan« hat mit dem barbarischen Leinwandhelden herzlich wenig zu tun. Die Handlung wirkt etwas an den Haaren herbeigezogen, und wer eine auffallende Ähnlichkeit zwischen Arnold »Conan« Schwarzenegger und dem mickrigen Bildschirm-Männlein feststellt, muß schon eine reichlich rege Phantasie haben. Selbst wenn man solche Hüpfspiele mag, überkommt einen rasch das große Gähnen. Da greife man lieber zum guten alten »Bruce Lee«, der viel mehr Spielspaß bietet.

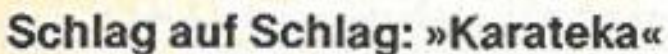
Evergreen-Tips für die Sammlung

Überhaupt gibt es dank der Kompatibilität zur 400/800-Serie einige betagte Klassiker, die auch heute noch spielenswert sind. Wer seinen Atari noch nicht allzulang hat und verzweifelt nach neuen Spielen sucht, sollte sich mal nach Evergreens wie »M.U.L.E.«, »Archon« oder »Dimension X« umhören. Letzteres gehört zu den wenigen Atari-Programmen, die nicht für den C 64 umgesetzt wurden. Einfacher Grund: Das Schießspiel, das im wesentlichen von seiner schnellen Grafik lebt, wäre in der Commodore-Version viel zu langsam geworden. Und wer sich nach der guten alten Computer-Steinzeit sehnt, kann sich immer noch den Original-»Pac Man« zulegen, der vor Jahren Ataris Superhit für das Videospiel-System 2600 war. Es werden aber hier und da noch Erfolgstitel anderer Computer nachträglich umgesetzt. So gibt es jetzt auch eine Atari-Version von Broderbunds langsamem, aber stilvollem Sport-Actionspiel »Karateka«, die sich hinter dem Apple II-Original nicht verstecken braucht.

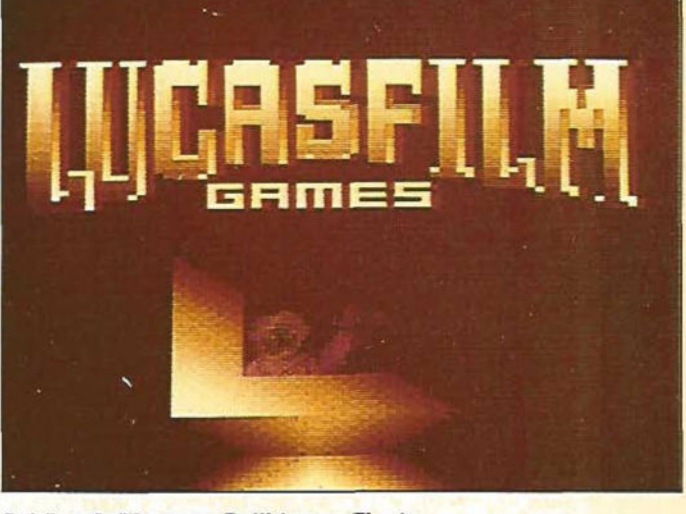
Ataris 8-Bit-Computer gehören sicher zu den mittelfristig auslaufenden Modellen auf dem Markt: Der 800XL wird zu Schleuderpreisen ausverkauft, und der neue 130XE verzeichnet nur mäßige Absatzzahlen. Angesichts dieser Situation ist auch kaum mit einem wesentlichen Aufschwung beim Spiele-Angebot zu rechnen. Auf der anderen Seite dürfte der Nachschub von heute auf morgen keinesfalls völlig einschlafen, da vor allem in Deutschland und den USA noch größere Stückzahlen auf dem Markt sind. Wer aber in den nächsten Jahren in einem üppigen Software-Angebot schwelgen will, sollte langsam einen Umstieg einplanen. Doch das Angebot an guten Atari-Spielen ist so groß, daß man ein ganzes Weilchen braucht, um sich durchzuspielen.

Zu guter Letzt noch ein Titel, der uns erst kurz nach Redaktionsschluß erreichte und den wir deshalb leider ohne Bild vorstellen müssen: »Mercenary« von »Encounter«-Autor Paul Woakes. Das Spiel ist eine einfallsreiche





Schlag auf Schlag: »Karateka«



Schöne Grüße vom »Ballblazer«-Finale

Neuer Hit mit 3D-Grafik

Mischung aus Flugsimulator, Geschicklichkeit, Strategie und Action und ist zu recht zivilen Preisen (42 Mark Kassette, 49 Mark Diskette) erhältlich. Das Glanzstück von »Mercenary« sind die vielen 3D-Vektorgrafiken, die diesen komplexen Spielgenuß zu einem garantierten Renner für 1986 werden lassen.

(Heinrich Lenhardt)

Bezugsquelle:
Compy-Shop, Gneisenaustr. 29, 4330 Mülheim/Ruhr, Tel. (0208) 497169 (USA-Importe)
Funtastic, Tannhäuserplatz 22, 8000 München 80, Tel. (089) 939894
Teledienst, Mainzer-Tor-Anlage 45, 6360 Friedberg, Tel. (06031) 91650

Cartridge-Experimente leichtgemacht

»CES« hat nichts mit der bekannten Messe in Las Vegas zu tun. Hier handelt es sich um ein Cartridge-Experimentier-System zum Selberbasteln.

Es gibt verschiedene Wege, den Atari-Computer mit Software zu versorgen. Programme oder Daten können einerseits manuell über die Tastatur in den Arbeitsspeicher des Computers gelangen. Das ist aber, wie das Einlesen von Daten mit dem Kassettenrecorder, ein recht langwieriger Vorgang. Mit einer Diskettenstation geht es schon einige Größenordnungen schneller.

Die maximale Geschwindigkeit ist bei Verwendung eines ROM-Cartridges zu erreichen, da hier dem Mikroprozessor ein direkter Datenzugriff erlaubt ist. Als weiterer Vorteil wäre noch das problemlose Austauschen der Cartridges zu nennen. Ebenso erfreulich ist, daß nach dem Ausschalten des Computers die Informationen in den ROMs erhalten bleiben.

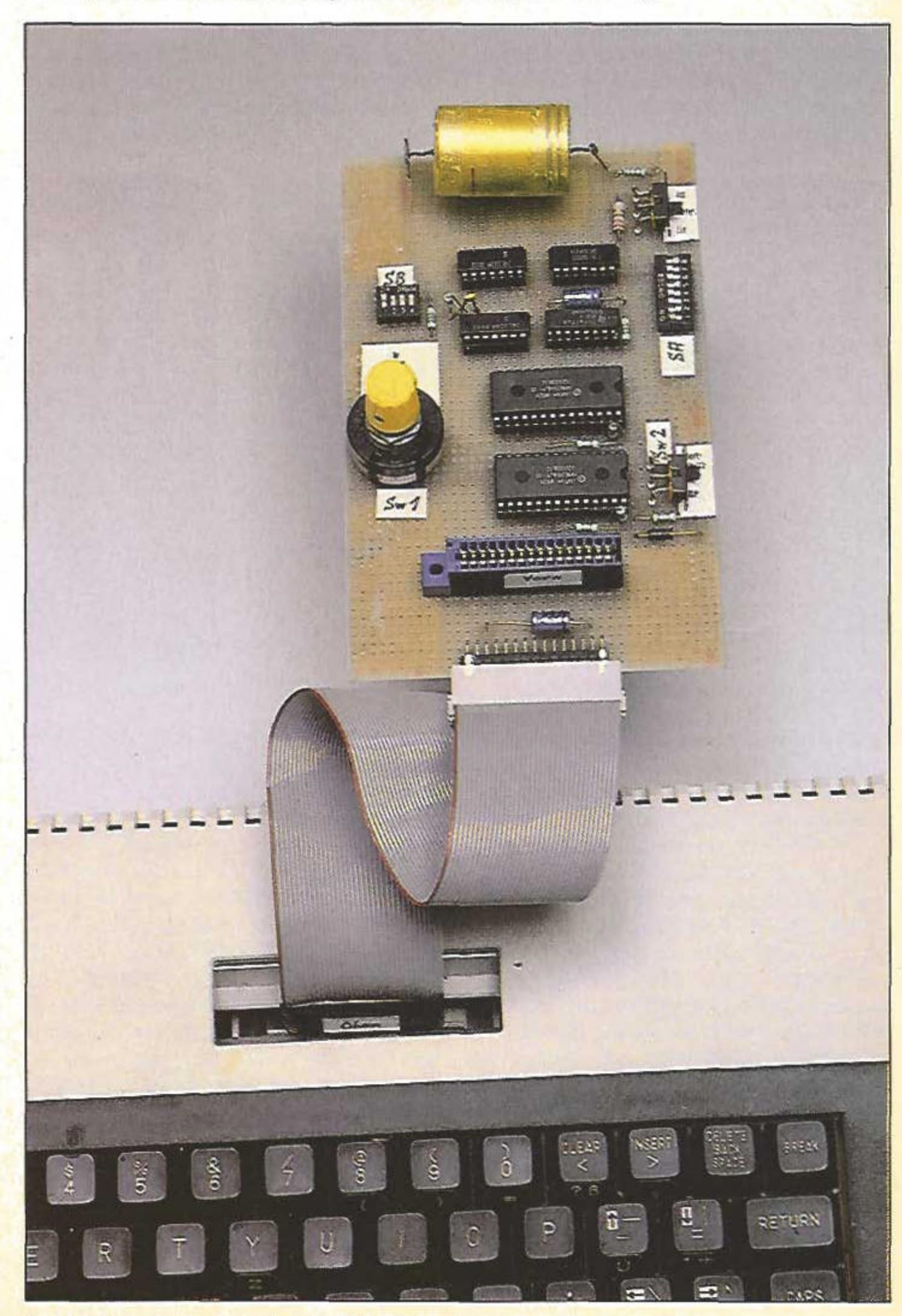
Beim Atari 800XL/130XE ist für den Anschluß solcher Cartridges ein sogenannter Cartridge-Slot vorgesehen. Er befindet sich bei Computern der XL-Serie auf der Oberseite des Gerätes, beim 130XE an der Rückseite. In der Anschlußbelegung unterscheiden sie sich jedoch nicht.

An Software auf Cartridge sind sowohl Spiele als auch Programmiersprachen erhältlich. Je nach Programm werden 8KByte oder 16KByte-Cartridges eingesetzt. Diese ROM-Cartridges fügen sich, wie aus Bild 1 ersichtlich ist, in den Speicheradreßraum des Computers ein.

Ein Cartridge mit einem eigenen Programm war vielleich schon hin und wieder der Wunsch eines Atari-Besitzers. Oder wäre es nicht auch erfreulich, die Software eines käuflich erworbenen Cartridges den eigenen Bedürfnissen anpassen beziehungsweise erweitern zu können? Dieser Wunsch als Vater des Gedankens war der Anstoß für die Entwicklung der in Bild 3 gezeigten Schaltung.

Das CES (Cartridge-Experimentier-System) befindet sich auf einer Platine im Europaformat (100 mm x 160 mm). Es wird über eine Flachbandleitung und einen Cartridge-Slot-Stecker an den normalen Modulschacht des Atari-Computers angeschlossen. Die Platine des CES weist ebenfalls ein Slot zur Aufnahme der üblichen Cartridges auf. Ein Wahlschalter (SW 1) ermöglicht die Entscheidung, mit dem Cartridge der Platine oder mit CES zu arbeiten. Wird kein Cartridge benötigt, so muß es nicht extra entfernt werden. Eine eigene Funktion erlaubt die Trennung von CES beziehungsweise Cartridge und Computersystem.

Mit dem CES lassen sich verschiedene Cartridge-Größen simulieren. Es sind sowohl 8-KByte- als auch 16-KByte-Nachbildungen realisierbar. Für die Datenspeicherung sind zwei statische 8-KByte-RAM-Bausteine verantwortlich, die jederzeit eine Änderung der Cartridge-Software erlauben.



So sieht die fertig aufgebaute Schaltung aus

Die Schreibfreigabe-Leitung (WE) der RAM-Bausteine liegt normalerweise auf High-Pegel, so daß nur das Lesen der Module gestattet ist. Zur Programmierung des CES kann man jedoch mit Hilfe des Schalters SW 2 auf RAM-Betriebsart umschalten (Stellung »RAM«). Das CES verhält sich dann wie eine ganz gewöhnliche RAM-Erweiterung. Die Schreibfreigabe kann aber auch softwaregesteuert erfolgen (SW 2 in Stellung »Softswitch«). Dazu jedoch später mehr.

Zur Einstellung der Betriebsart des CES (Größe, Lage im Speicheradreßraum etc.) dienen zwei DIP-Schalter SA1 bis SA8 und SB1 bis SB4.

Zunächst einiges zu den am Cartridge-Slot des Atari angebrachten Anschlußleitungen. Da wäre einerseits der vollständige Datenbus, die Adreßleitungen A0 bis A12 sowie einige Steuerleitungen und eine 5-Volt-Spannungs versorgung.

Mit dem 13-Bit-Adreßbus läßt sich ein Speicheradreßraum von 8 KByte ansprechen. Über die Meldeleitungen M4 und M5 (Aktiv High) wird dem Computer mitgeteilt, in welchem Speicherbereich des Systems sich ein Cartridge befindet. Die Speicherverwaltungslogik im Atari weiß dann, an welchen Stellen sie das interne RAM ausblenden und das Cartridge einfügen soll. Dabei bedeutet:

M4 = High - 8 KByte Cartridge im Bereich zwischen \$8000 bis \$9FFF
M5 = High - 8 KByte Cartridge im Bereich zwischen \$A000 bis \$BFFF

Liegen beide Meldeleitungen auf High, so entspricht das einem 16K-Cartridge im Adreßbereich von \$8000 bis \$BFFF.

Die Steuerleitungen S4 und S5 sind Low-Aktiv und kommen vom Computer. Sie dienen dem Chip-Select (Baustein-Auswahl) der ROMs im Cartridge. S4 wird Low, wenn ein Zugriff auf den Speicherbereich zwischen \$8000 bis \$9FFF erfolgt. Bei einem Zugriff auf den Bereich von \$A000 bis \$BFFF wird S5 Low.

Diese Steuer- und Meldeleitungen werden über den Wahlschalter SW 1 entweder mit dem Cartridge-Slot auf der Platine des CES oder mit den entsprechenden Anschlußpunkten der CES-Logik verbunden; abhängig davon, ob ein normales Cartridge im Slot steckt oder das CES betrieben werden soll. Befindet sich SW 1 in Mittelstellung, laufen die Steuer- und Meldeleitungen leer. Der Computer registriert dann, daß kein Cartridge angeschlossen ist.

Eine weitere Steuerleitung am Cartridge-Slot des Atari ist die

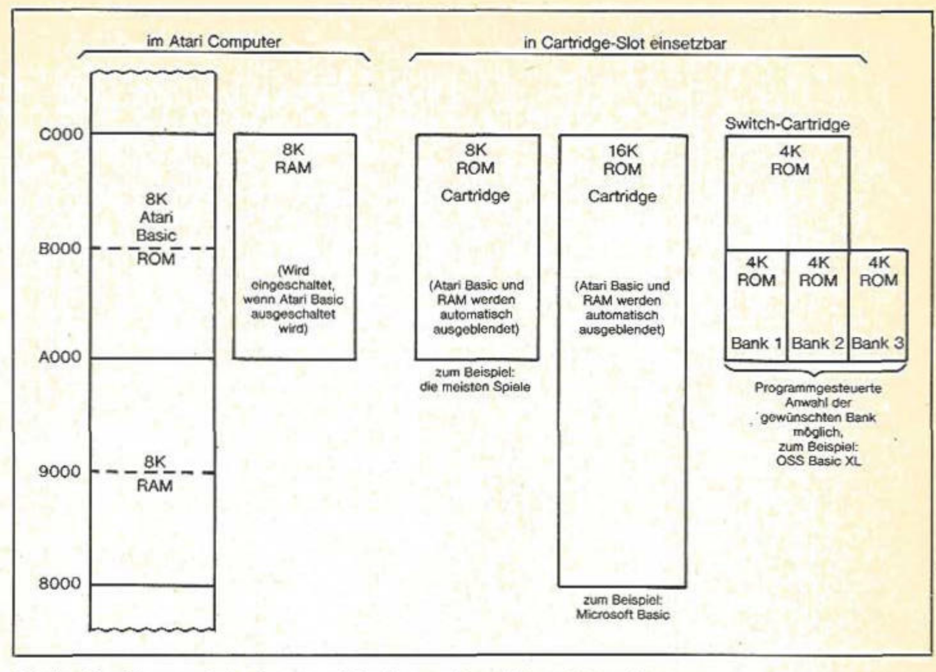


Bild 1. So fügen sich die verschiedenen Cartridge-Arten in den Speicheradreßraum der Atari-Computer ein

Schreibfreigabe-Leitung WE. Sie steht immer dann auf Low, wenn in eine Speicherstelle geschrieben wird. Die Schreibfreigabe-Signale werden nun beim CES über eine Tor-Schaltung an die WE-Anschlüsse der RAMs geleitet. Damit ist es möglich, ein Beschreiben derjenigen RAMs, die zum CES gehören, zu verhindern (ROM-Betrieb).

Die Anschlüsse des Cartridge-Slot am Atari reichen schon aus, um ein normales Cartridge mit dem CES zu simulieren. Mit den beiden noch verbleibenden Anschlüssen PHI2 und CCTL lassen sich noch einige Zusatzfunktionen realisieren.

Am Anschluß PHI2 steht der Systemtakt des Atari zur Verfügung, von dem alle anderen Speichersteuersignale abgeleitet werden (zum Beispiel Schreibfreigabe). Besondere Bedeutung kommt jedoch dem Anschluß CCTL zu. Diese Leitung wird aktiv Low, wenn ein Zugriff auf den Speicheradreßraum \$D500 bis \$D5FF erfolgt.

Im Atari ist dieser Speicherbereich weder durch das RAM noch vom ROM belegt. Die CCTL-Leitung wird nun beim CES in Verbindung mit den untersten 6 Adreßbusleitungen und dem Systemtakt PHI2 zur Software-Steuerung des CES benutzt. Mit Hilfe von IC 3 (6 Bit Dual-Flipflop) können die Zustände der Adreßleitungen A0 bis A5 bei einem Zugriff auf den Speicherbereich \$D500 bis \$D5FF gelegt werden. Diese gespeicherten Zustände werden nun zur Steuerung des CES benutzt. Ein Beispiel soll dies verdeutlichen:

Soll ein Zugriff auf die Speicherstelle \$D510 erfolgen, so sind die unteren 6 Adreßleitungen zum Zeitpunkt des Zugriffs (Lesen oder Schreiben!) folgendermaßen gesetzt:

A0 bis A3 = Low
A4 = High
A5 = Low

Dieser Zustand wird nun in IC 3 gespeichert und liegt auch so lange an dessen Ausgängen Q0 bis Q5, bis eine andere Adresse im Bereich zwischen \$D500 bis \$D5FF angesprochen wird. So läßt sich dann ein neuer Zustand speichern. In unserem Beispiel liegt also an Ausgang Q4 ein High Pegel an.

Cartridge- Unterseite			Oberseite (*)
S4	1	A	M4
A3	2	B	GND
A2	3	C	A4
A1	4	D	A5
A0	5	E	A6
D4	6	F	A7
D5	7	H	A8
D2	8	J	A9
D1	9	K	A12
D0	10	L	D3
D6	11	M	D7
S5	12	N	A11
+5V	13	P	A10
M5	14	R	WE
CCTL	15	S	PHI2

(*) Zeigt bei der XL-Serie zur Tastatur

Bild 2. Die Anschlußbelegung des Atari Cartridge-Slot

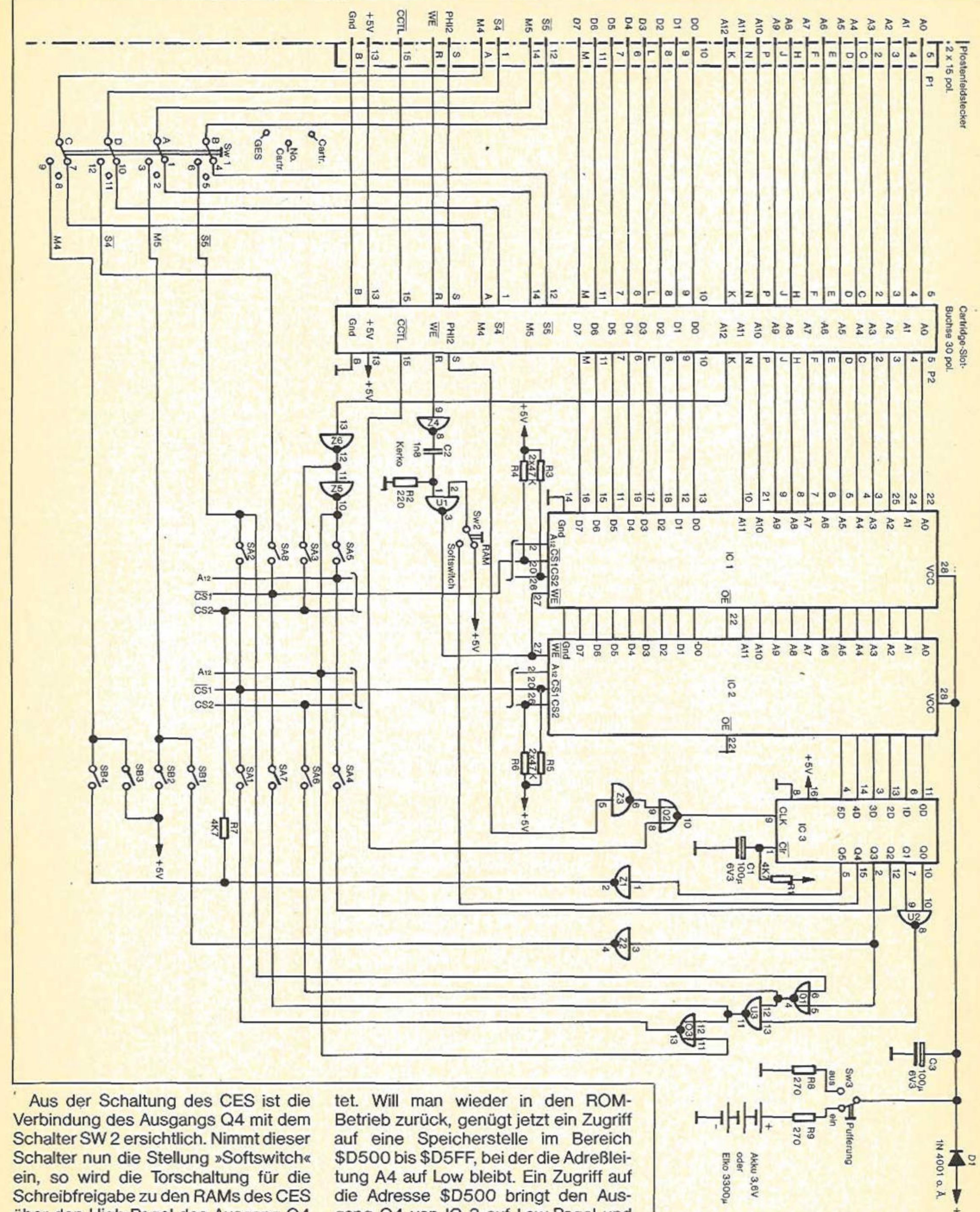


Bild 3. Schaltplan zu CES (Cartridge-Experimentier-System)

Aus der Schaltung des CES ist die Verbindung des Ausgangs Q4 mit dem Schalter SW 2 ersichtlich. Nimmt dieser Schalter nun die Stellung »Softswitch« ein, so wird die Torschaltung für die Schreibfreigabe zu den RAMs des CES über den High-Pegel des Ausgang Q4 von IC 3 geöffnet und somit über die Adreßleitung A4 gesteuert! In unserem Beispiel wird also, sobald sich SW 2 in Stellung »Softswitch« befindet, das CES in RAM-Betriebsart geschaltet. Will man wieder in den ROM-Betrieb zurück, genügt jetzt ein Zugriff auf eine Speicherstelle im Bereich $D500 bis $D5FF, bei der die Adreßleitung A4 auf Low bleibt. Ein Zugriff auf die Adresse $D500 bringt den Ausgang Q4 von IC 3 auf Low-Pegel und sperrt damit wieder die Torschaltung. Das RC-Glied R1 bis C1 am Clr-Eingang von IC 3 sorgt dafür, daß nach dem Einschalten des Computers und des CES alle Ausgänge auf Low gesetzt sind.

Integrierte Schaltkreise	
IC1, IC2	HM6264LP-1S (oder ähnlich)
IC3	74LS174
(Z1 bis Z6) = IC4	74LS04
(U1 bis U3) = IC5	74LS00
(O1 bis O3) = IC6	74LS02
Schalter	
SW1	Drehschalter, drei Stellungen mit vier Schaltebenen
SW2, SW3	Schiebeschalter 1xUM
SA1 bis SA8	DIP-Schalter, 8xEin
SB1 bis SB4	DIP-Schalter, 4xEin
Widerstände	
R1, R7	4,7 KOhm
R2	220 Ohm
R3, R4, R5, R6	47 KOhm
R8, R9	270 Ohm
Kondensatoren	
C1, C3	100 µ, 6V3 (Elko)
C2	1n8 (Keramik-Elko)
Sonstiges:	
Zur Pufferung wahlweise ein 3.6-Volt-Akku oder ein 3300-µFarad-Elko	

Bild 4. Bauteileliste zum Cartridge-Experimentier-System

Im CES werden für die Datenspeicherung statische CMOS-RAMs des Typs 6264LP-15 verwendet, die im nicht selektierten Zustand, also im Standby-Betrieb, eine sehr geringe Stromaufnahme haben. Deshalb ist in der Schaltung eine Pufferung der Spannungsversorgung für die RAM-Bausteine vorgesehen. So bleiben auch nach Abschalten der Computerspannungsversorgung die Daten im CES für längere Zeit erhalten. Man muß nicht unbedingt einen NC-Akku zur Pufferung einsetzen. Ein Elko mit 3300 Mikro-Farad im CES erfüllt für immerhin ganze 11 Stunden denselben Zweck. Soviel zur Arbeitsweise der Schaltung.

Wichtig: Soll die Einstellung der DIP-Schalter des CES verändert werden, so müssen zuvor Computer und Pufferung des CES ausgeschaltet werden!

Nun einige Erläuterungen zu den Betriebsarten, wie sie in Tabelle 1 aufgeführt sind.

8-KByte-Cartridges im Bereich zwischen $8000 und $9FFF

In dieser Betriebsart steht das CES direkt im Anschluß an den Adreßraum des eingebauten Basic zur Verfügung. Man kann dort zum Beispiel sehr einfach oft benötigte Hilfsprogramme unterbringen. Ist SW 2 in Stellung »RAM«, so kann man das CES von Basic aus wie ein RAM betreiben. In Stellung »Softswitch« erfolgt eine softwaremäßige Umschaltung zwischen RAM- und ROM-Betrieb. Wird die Speicherstelle 54544 dezimal (zum Beispiel mit »POKE 54544,1«) angesprochen, so wird der RAM-Betrieb eingeschaltet. Mit »POKE 54528,1 befindet man sich wieder im ROM-Betrieb.

DIP-Schalter SA								DIP-Schalter SB				Betriebsart des CES
1	2	3	4	5	6	7	8	1	2	3	4	
0	0	0	0	0	0	0	1	0	0	1	0	**8K-Cartridge** - Bereich $8000 - $9FFF
0	0	0	0	0	0	0	1	0	0	0	1	- Bereich $8000 - $9FFF (softwaregesteuert) (ausblendbar)
0	1	0	0	0	0	0	0	0	1	0	0	- Bereich $A000 - $BFFF
0	1	0	0	0	0	0	1	0	1	1	0	**16K-Cartridge** - Bereich $8000 -$BFFF
1	0	1	1	0	1	1	0	1	0	0	0	**16K-Switch-Cartridge** - Bereich $A000 - $BFFF (mit Bank-Switching)
1 = Schalter geschlossen								0 = Schalter geöffnet				

▲ Tabelle 1. Die Betriebsarten des CES

```
1000 ;--------------------------------
1010 ;              TRANSCAR
1020 ;
1030 ; Ermoeglicht das Auslesen von
1040 ; Daten aus dem im Slot befind-
1050 ; lichen Cartridge bzw. das Kopie-
1060 ; ren von Daten aus dem RAM in das
1070 ; Cartridge-Experimentier-System.
1080 ;
1090 ;
1100 ; SRCBOT = Anfangsadr. bzw Endadr.
1110 ; SRCTOP   des Quellspeicherber.
1120 ; DSTBOT = Anfangsadr. des Ziel-
1130 ;          speicherbereichs
1140 ;
1150 ;--------------------------------
1160 ;
1170 TEMP =  $D0      ;Zwischenspeicher
1180 ;
1190 SRCBOT = $A000
1200 SRCTOP = $BFFF
1210 DSTBOT = $6000
1220 ;
1230 CONSOL = $D01F  ;CONSOL-Tasten
1240 NMIEN = $D40E   ;NMI-ENABLE-Ctrl.
1250 ;
1260 RAMCAR = $D510  ;Adr. f. Schreib-
1270 ;               ;freigabe (je nach
1280 ;               ;Betriebsart des
1290 ;               ;CES)
1300 ROMCAR = $D500  ;Adr. f. Schreib-
1310 ;               ;sperre   (je nach
1320 ;               ;Betriebsart des
1330 ;               ;CES)
1340 ;
1350      *=  $0600
1360 ;
1370 SWSTRT
1380      LDA NMIEN   ;Alle Interrupts
1390      STA TEMP    ;sperren!
1400      LDA #0
1410      STA NMIEN
1420 ;
1430 WTSTRT
1440      LDA CONSOL  ;Warten bis die
1450      CMP #6      ;'START'-Taste
1460      BNE WTSTRT  ;gedrueckt wurde.
1470 ;
1480 ENWRT
1490      STA RAMCAR  ;Schreibfreigabe
1500 ;               ;fuer CES
1510 ;
1520 ;
1530 TRANS
1540      LDX # <SRCTOP
1550      LDY # >SRCTOP
1560 VON
1570      LDA SRCBOT  ;Kopiere Daten von
1580 NACH ;           ;SRCBOT bis SRCTOP
1590      STA DSTBOT  ;nach DSTBOT ff.
1600      CPY VON+2
1610      BNE INCVON
1620      CPX VON+1
1630      BEQ RETURN
1640 INCVON
1650      INC VON+1
1660      BNE INCTO
1670      INC VON+2
1680 INCTO
1690      INC NACH+1
1700      BNE VON
1710      INC NACH+2
1720      CLC
1730      BCC VON
1740 ;
1750 DISWRT
1760      STA ROMCAR  ;CES auf ROM-Be-
1770 ;               ;trieb schalten.
1780 ;
1790 RETURN
1800      LDA TEMP    ;Interrupts frei-
1810      STA NMIEN   ;geben.
1820      RTS
1830 ;
1840      *=  $02E0   ;Autostartfile
1850 ASTRT ;          erzeugen!
1860      .WORD SWSTRT
1870 ;
```

◀ Listing 1. Quelltext zum Lesen des Inhalts eines Cartridges (MAC/65)

8-KByte-Cartridge im Bereich zwischen $8000 bis $9FFF; softwaregesteuert und ausblendbar

Diese Betriebsart erlaubt es, das CES per Programm außer Betrieb zu setzen und das darunterliegende RAM des Atari zu aktivieren. So erhält man im Adreßraum $8000 bis $9FFF zwei 8 KByte große Speicherbereiche. Einmal das RAM des Computers, dessen Inhalt nach Ausschalten des Atari verlorengeht; zum anderen das CES, dessen Inhalt durch die Pufferung mittels Akku oder Elko auch nach Abschalten des Computers erhalten bleibt.

Folgende Zustände sind zum Beispiel von Basic aus steuerbar:

»POKE 54528,1« - CES als ROM aktiviert (=Einschaltzustand)

»POKE 54544,1« - CES als RAM aktiviert

»POKE 54560,1« - RAM des Computers aktiviert.

8-KByte-Cartridge im Bereich zwischen $A000 bis $BFFF

Dies ist wohl die am häufigsten anzutreffende Cartridge-Version für Atari-Computer. Die meisten Cartridge-Spiele liegen in diesem Speicheradreßbereich. Aber auch das Atari-Cartridge-Basic, wie es früher für die Atari 400 und 800 Computer geliefert wurde, befindet sich in diesem Bereich.

Vom eingebauten Basic der XL- und XE-Serie ist das CES in dieser Betriebsart nicht mehr zu kontrollieren. Das eingebaute Basic liegt nämlich ebenfalls im gleichen Adreßraum und ist deshalb bei eingeschaltetem CES desaktiviert.

Über die Maschinensprache ist jedoch ein Zugriff auf den Inhalt des CES möglich. Bei Listing 1 handelt es sich um ein solches Umschaltprogramm. Mit einem Assembler, wie zum Beispiel MAC/65, kann man Maschinenspracheprogramme für jeden beliebigen Adreßraum schreiben, assemblieren und dann auf Disk speichern. Ein solches Programmfile wird dann

```
1000 ;-------------------------------------
1010 ;              SWITCHER
1020 ;
1030 ;Ermoeglicht das Umschalten vom
1040 ;CES auf das im Slot des CES
1050 ;befindl. Cartridge und umgekehrt.
1060 ;
1070 ;Nach Programmstart 'haengt' der
1080 ;Computer und es kann mit dem
1090 ;Wahlschalter Sw 1 auf der Platine
1100 ;des CES umgeschaltet werden.
1110 ;Mit Druck auf die 'START'-Taste
1120 ;wird der Atari wieder 'in Gang'
1130 ;gesetzt und das DOS meldet sich.
1140 ;
1150 ;-------------------------------------
1160 ;
1170 TEMP =  $D0        ;Zwischenspeicher
1180 NCART = $C4C9      ;CARTCK berechnen
1190 CONSOL = $D01F     ;CONSOL-Tasten
1200 NMIEN = $D40E      ;NMI-ENABLE-Ctrl.
1210 ;
1220      *=  $0600
1230 ;
1240 SWITCH
1250      LDA NMIEN     ;Alle Interrupts
1260      STA TEMP      ;sperren!
1270      LDA #0
1280      STA NMIEN
1290 WTSTRT
1300      LDA CONSOL    ;Warten bis die
1310      CMP #6        ;'START'-Taste
1320      BNE WTSTRT    ;gedrueckt wurde.
1330 NEUCAR
1340      JSR NCART     ;Neue Cart.-Check-
1350 ;                   summe berechnen.
1360 RETURN
1370      LDA TEMP      ;Interrupts frei-
1380      STA NMIEN     ;geben.
1390      RTS
1400 ;
1410 ASTRT
1420      *=  $02E0
1430      .WORD SWITCH ;Autostart-File
1440 ;                   erzeugen!
```

Listing 2. Dieses Assembler-Programm dient zum Umschalten zwischen dem CES und einem eingesteckten ROM-Modul

beim Laden vom DOS automatisch in den dafür vorgesehenen Speicherraum geschrieben.

Es muß also im Prinzip folgenden Aufbau haben, damit es in den Speicher des CES geladen werden kann:

- Lade einen Wert, beispielsweise 0, nach $D510 (RAM-Betriebsart des CES).
- Lade die Daten für den Adreßraum $A000 bis $BFFF in das CES.
- Lade einen weiteren Wert nach $D500 (ROM-Betriebsart des CES).

Die Vorgehensweise für das Laden von Daten in das CES geschieht folgendermaßen:

1. Den Computer und die Pufferung des CES ausschalten.
2. DIP-Schalter für die entsprechende Betriebsart einstellen.
3. Schalter SW 1 auf Stellung »CES«.
4. Computersystem booten und Pufferung des CES einschalten.
5. Das File mit oben beschriebenem Aufbau vom DOS aus laden.

Die über DOS geladenen Daten befinden sich nun im CES.

16-KByte-Cartridge im Adreßbereich zwischen $8000 bis $BFFF.

Hierbei handelt es sich um zwei 8-KByte-Cartridges, die einen zusammenhängenden Adreßraum des Computers überdecken. Eine Kontrolle von Basic aus ist auch hier dadurch verhindert, daß das Basic von der oberen Hälfte des 16-KByte-Bereichs überlagert wird und somit nicht zugänglich ist. Es bleibt nur noch der Weg über die Maschinensprachebene.

16-KByte-Switch-Cartridge im Adreßbereich zwischen $A000 bis $BFFF

Wie aus Bild 1 ersichtlich handelt es sich hier um eine etwas komplexere Betriebsart. Das CES bietet die volle Speicherkapazität von 16 KByte, belegt aber nur 8 KByte Adreßraum im Computer. Dies macht das sogenannte Bank-Switching möglich. Der Bereich von $B000 bis $BFFF ist normalerweise immer eingeblendet. Ein dort befindliches Maschinenprogramm kann nun durch Zugriff auf bestimmte Speicherstellen zwischen $D500 bis $D5FF einen der drei Speicherbänke im Bereich $A000 bis $AFFF aktivieren und zu einem dort befindlichen Unterprogramm verzweigen. Nach Abarbeitung des Unterprogramms erfolgt der Rücksprung in das aufrufende Programm im Bereich von $B000 bis $BFFF. Jetzt kann noch eine andere Bank eingeblendet werden, um eventuell ein dort vorliegendes Unterprogramm auszuführen.

Mit Hilfe dieser Technik passen 16-KByte-Module in nur 8-KByte-Speicher! Die einzelnen Speicherbänke werden durch Zugriff auf folgende Adressen aktiviert:

Adresse	Bank
$D5X0	Bank 1 (untere 4 KByte in IC 1)
$D5X3	Bank 2 (untere 4 KByte in IC 2)
$D5X4	Bank 3 (obere 4 KByte in IC 1)
X=0	> ROM-Betriebsart
X=1	> RAM-Betriebsart

Nach diesem Prinzip arbeiten übrigens auch die Super-Cartridges von OSS. Will man Software in dieser Betriebsart ins Cartridge laden, so ist schon während des Ladevorgangs von Disk die Hardware des CES auf die entsprechenden Speicherstellen im Bereich ab $D5XX einzustellen. Ein Programmfile sollte also folgenden Aufbau haben:

1. Lade einen Wert, beispielsweise 0, nach $D510 (entspricht RAM-Betrieb des CES und Bank 1 ist eingeschaltet).
2. Lade die Daten für Bank 1 nach $A000 bis $AFFF und unmittelbar danach den Bereich von $B000 bis $BFFF.
3. Lade einen Wert nach $D513 (entspricht dem RAM-Betrieb des CES. Bank 2 ist eingeschaltet).
4. Lade die Daten für Bank 2 nach $A000 bis $AFFF.
5. Lade einen Wert nach $D514 (entspricht RAM-Betrieb des CES. Bank 3 ist eingeschaltet).
6. Lade die Daten für Bank 3 nach $A000 bis $AFFF.
7. Lade einen Wert nach $D500 (entspricht ROM-Betrieb des CES. Bank 1 ist eingeschaltet).

Das Programm befindet sich nun im CES und kann aufgerufen werden.

Soviel an Hinweisen zum Gebrauch des CES (für Anwender), die eigene Cartridge-Programme schreiben wollen. Es würde den Rahmen sprengen, hier noch auf Details einzugehen, wie zum Beispiel das Betriebssystem die verschiedenen Cartridgearten behandelt, wann ein Cartridge-Selbststart erfolgt und wie das Betriebssystem registriert, ob ein Cartridgewechsel erfolgt ist. Mehr dazu kann man der entsprechenden Literatur entnehmen (zum Beispiel DeReAtari, Atari 600 XL/800XL inter von Data Becker) oder durch Experimentieren herausfinden.

CES erlaubt also, wie schon zu Beginn bemerkt, den Inhalt von normalen Cartridges herauszufinden. Im Prinzip geht das folgendermaßen vor sich:

- Im Cartridge-Slot der CES-Platine befindet sich das zu untersuchende Cartridge.
- Das CES wird auf die gleiche Betriebsart wie das Cartridge eingestellt.
- Mittels eines kleinen Maschinenprogramms wird der Inhalt des Cartridges ins RAM kopiert.
- Anschließend wird auf CES-Betrieb umgeschaltet, und die im RAM vorliegenden Daten werden, ebenfalls mit einem kleinen Maschinenprogramm, ins CES verlagert.

Der Inhalt des CES kann nun mit einem Monitorprogramm untersucht und disassembliert werden. Schaltet man das CES auf RAM-Betrieb, so lassen sich auch Änderungen vornehmen. Ebenso besteht die Möglichkeit, den Inhalt des CES auf Diskette zu speichern und später wieder zu laden.

Das in Listing 2 abgedruckte Programm erledigt den Datentransfer zwischen Cartridge und RAM, zum Beispiel für ein 8-KByte-Cartridge im Speicherbereich zwischen $A000 bis $BFFF. Eine Änderung dieses Bereiches läßt sich recht einfach realisieren. Das Programm kopiert die Daten aus dem Speicherbereich zwischen SRCBOT und SRCTOP (im Listing beispielsweise $A000 bis $BFFF) in den Speicherbereich ab DSTBOT (im Listing beispielsweise $6000 bis $7FFF). Dieses Programm wird assembliert und unter dem Namen »READOUT.COM« auf Diskette gespeichert.

Dann wird das Programm noch einmal assembliert, jedoch mit den Adressen $6000 für SRCBOT, $7FFF für SRCTOP und $A000 für DSTBOT und mit dem Namen »WRITECES.COM« auch noch auf Diskette gespeichert. Und so wird's gemacht:

1. Computersystem und CES-Pufferung ausschalten.
2. Zu untersuchendes Cartridge in den Slot des CES stecken.
3. Betriebsart des CES entsprechend dem zu untersuchenden Cartridge einstellen.
4. Schalter SW 1 in Stellung »CES«.
5. Schalter SW 2 in Stellung »Softswitch«.
6. Computersystem und Pufferung des CES einschalten.
7. Bootprozeß abwarten, bis sich das DOS meldet.
8. Das Programm »READOUT.COM« vom DOS aufrufen.
9. Schalter SW 1 auf Stellung »Cart.« schalten.
10. START-Taste drücken und abwarten, bis sich das DOS wieder meldet.
11. Programm »WRITECES.COM« vom DOS aufrufen.
12. Schalter SW 1 auf Stellung »CES« schalten.
13. START-Taste drücken und warten, bis sich das DOS wieder meldet.

Im CES befinden sich nun die gleichen Daten wie in dem zu untersuchenden Cartridge. Jetzt läßt sich der Inhalt des CES beispielsweise mit einem Monitorprogramm wie Bug/65 untersuchen und disassemblieren.

Soviel zum Umgang mit dem CES. Wie der Name schon sagt, handelt es sich um ein Experimentiersystem. Beschäftigt man sich intensiv mit dem System, so wird man bestimmt noch andere, hier nicht aufgeführte Betriebsarten entdecken.

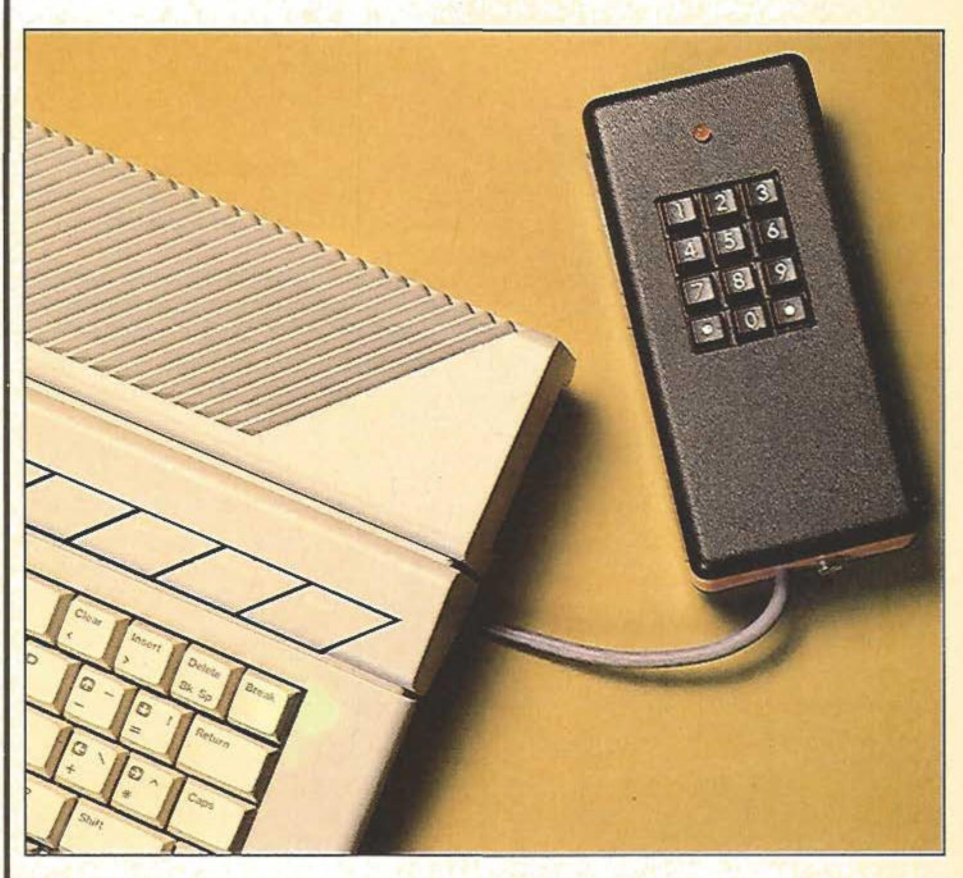
Zum Schluß noch einiges zum hardwaremäßigen Aufbau der CES-Platine. Wie bereits erwähnt, erfolgte die Realisierung der Schaltung mit einer 100 mm x 160 mm großen, einseitigen Lochrasterplatine mit Lötpunktraster. Die Verdrahtung ist in Fädeltechnik ausgeführt und deshalb schnell durchführbar. Zu einem späteren Zeitpunkt könnte man sich natürlich auch eine Platine entwerfen. Auch Änderungen lassen sich schnell und einfach vornehmen. Schließlich handelt es sich um ein Experimentiersystem.

Die nötigen Anschlüsse zum Cartridge-Slot des Atari-Computers sind auf Pfostenfeldstecker (2 x 15 Pole) geführt. Die Verbindung zum Cartridge-Slot des Atari übernimmt ein 30adriges Flachbandkabel. An der Seite zur CES-Platine ist eine 34polige Pfostenfeldbuchse in Schneidklemmtechnik angepreßt. An der Cartridge-Slot-Seite zum Atari ist das Flachbandkabel an eine doppelseitige Streifenleiterplatine angelötet und mit Zwei-Komponenten-Klebstoff fixiert.

(H. D. Jankowski/wb)

PIA – die dritte Schnittstelle

Mit sehr einfachen und preiswerten Mitteln läßt sich sogar eine selbstgebastelte Zusatztastatur an Ihren Joystick-Port anschließen.



▲ Die Zusatztastatur läßt sich problemlos in einem handlichen Gehäuse unterbringen.

PROGRAMM-STECKBRIEF	
Programmname	PIA-Steuerung
Programmtyp	Utility
Programmiersprache	Basic und Maschinensprache
Programmlänge	insgesamt 6272 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	laut Bauanleitung
Eingabehilfe	Prüfsummer
Bemerkung	In Verbindung mit der Zusatzschaltung läßt sich eine Tastatur oder ein Schaltinterface steuern
Leserservice	Diskette (PIA1.BAS/PIA2.BAS)

Das ganze Geheimnis der Joystick-Schnittstelle ist die sogenannte PIA. Es handelt sich hier um einen Standardperipheriebaustein des Typs 6520. Dieses Bauteil verfügt unter anderem über zwei 8 Bit breite Ports, die sich sowohl für die Einals auch für die Ausgabe von Daten nutzen lassen. Wie man diese Ports verwendet, ist dabei softwaremäßig völlig frei programmierbar. Leider macht sich hier jedoch ein Nachteil der Atari XL- und XE-Generation deutlich bemerkbar. Während die alten Modelle noch über vier Joystickports verfügten, sparte man bei den neueren Modellen zwei Ports ein. Durch diese Einschränkung konnte der zweite Port der PIA natürlich nicht mehr nach außen geführt werden. Besitzt man also ein neueres Gerät, so kann man nur noch mit einem der beiden PIA-Ports arbeiten und ver-

fügt statt über insgesamt 16 Datenbits nur mehr über 8. Selbst ein hardwaremäßiger Eingriff, bei dem man den fehlenden Port nach außen führen würde, ist nicht durchführbar. Dieser Port übernimmt nämlich nun andere Aufgaben, vor allem in der Zusammenarbeit mit der MMU (Memory Management Unit oder Speicherverwaltungseinheit).

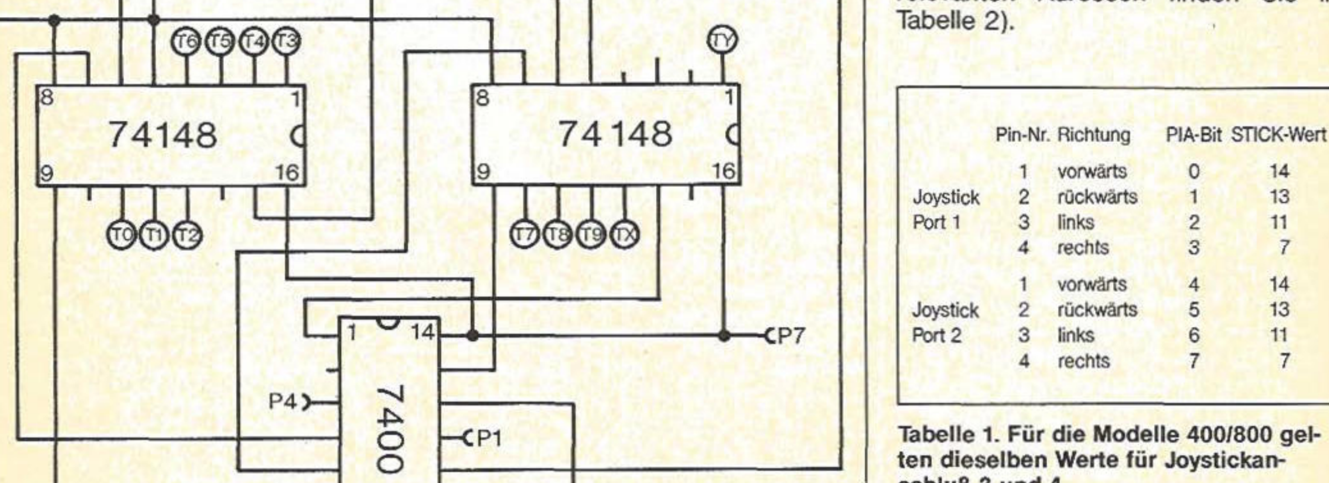
Dem Atari geht ein Licht auf

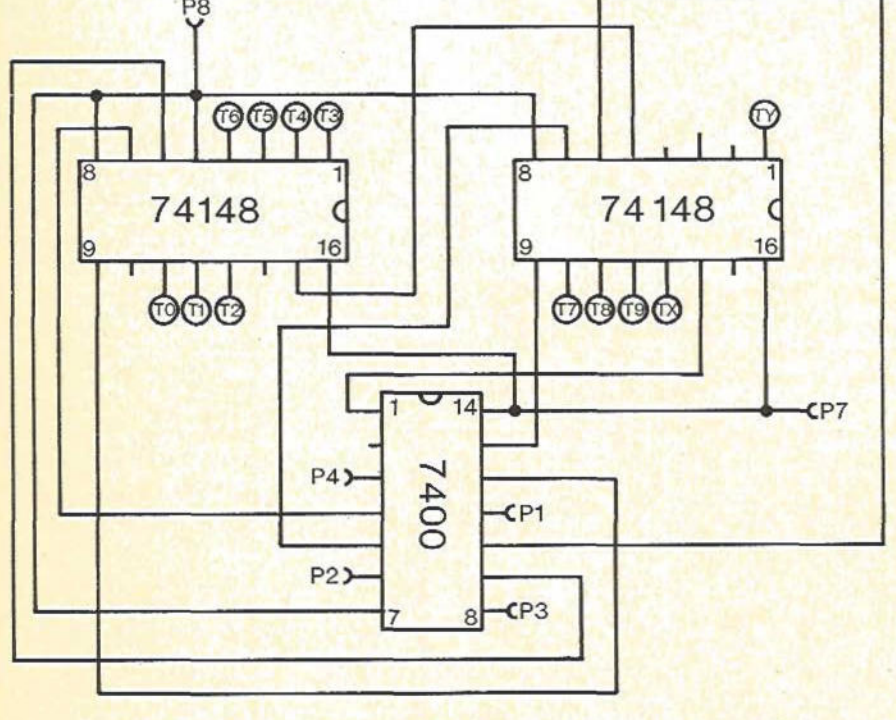
Im Normalzustand ist die PIA auf Eingabe programmiert. So ist es einsichtig, daß man bei der Auslenkung des Joysticks eigentlich nichts anderes tut, als einzelne Bits des Ports zu setzen oder zu löschen. Aus dieser Tatsache erklären sich auch die auf den ersten Blick etwas kuriosen Werte, die man mit Hilfe des STICK-Befehls erhält. Lenkt man den Joystick nicht aus, erhält man beispielsweise den Wert 15. Dies bedeutet, daß alle 4 für einen Joystick zuständigen Bits gesetzt sind (1111 binär ergibt 15 dezimal). Ist der Joystick nach oben gelenkt, so wird das niedrigste Bit gelöscht und man erhält den Wert 14 (1110 binär ergibt 14 dezimal). Welche Joystickbewegungen die einzelnen Bits beeinflussen, können Sie aus Tabelle 1 entnehmen.

Will man nun die Daten der PIA abrufen, erweist sich die STICK-Funktion nicht als besonders sinnvoll. Zum einen erfaßt man pro Befehl nur 4 Bit, zum anderen lassen sich mit diesem Befehl keine Daten an die PIA senden. Deshalb bedient man sich besser direkt der hierfür vorgesehenen Speicherstellen. Jedem der beiden PIA-Ports sind zwei Speicherstellen zugeordnet. Einmal das PIA-Register, das die zu sendenden oder zu empfangenden Daten enthält. Dann das Datenrichtungsregister, das festlegt, welche Bits des PIA-Registers für die Eingabe beziehungsweise Ausgabe verwendet werden (die dafür relevanten Adressen finden Sie in Tabelle 2).



Auf der abgebildeten Testplatine wurden die Schaltungen von Bild 2 und Bild 3 auf einer Lochrasterplatine realisiert





12	Taster, Typ Digitast
2	IC 74LS148
1	IC 74LS00
1	Stecker 9pol.
1	Kabel 6pol.
1	Gehäuse
1	Lochrasterplatine

Bauteileliste für eine Zusatztastatur

◀ **Bild 1. Schaltplan für eine Zusatztastatur**

	Pin-Nr.	Richtung	PIA-Bit	STICK-Wert
	1	vorwärts	0	14
Joystick	2	rückwärts	1	13
Port 1	3	links	2	11
	4	rechts	3	7
	1	vorwärts	4	14
Joystick	2	rückwärts	5	13
Port 2	3	links	6	11
	4	rechts	7	7

Tabelle 1. Für die Modelle 400/800 gelten dieselben Werte für Joystickanschluß 3 und 4

Port A	54016
Port B	54017
Port A DRR	54018
Port B DRR	54019
Joystick 1	632
Joystick 2	633
Joystick 3	634
Joystick 4	635

Tabelle 2. Der Zusatz »DRR« steht für »Datenrichtungsregister«. Die Adressen für Joystick 3 und 4 betreffen nur die alten Modelle.

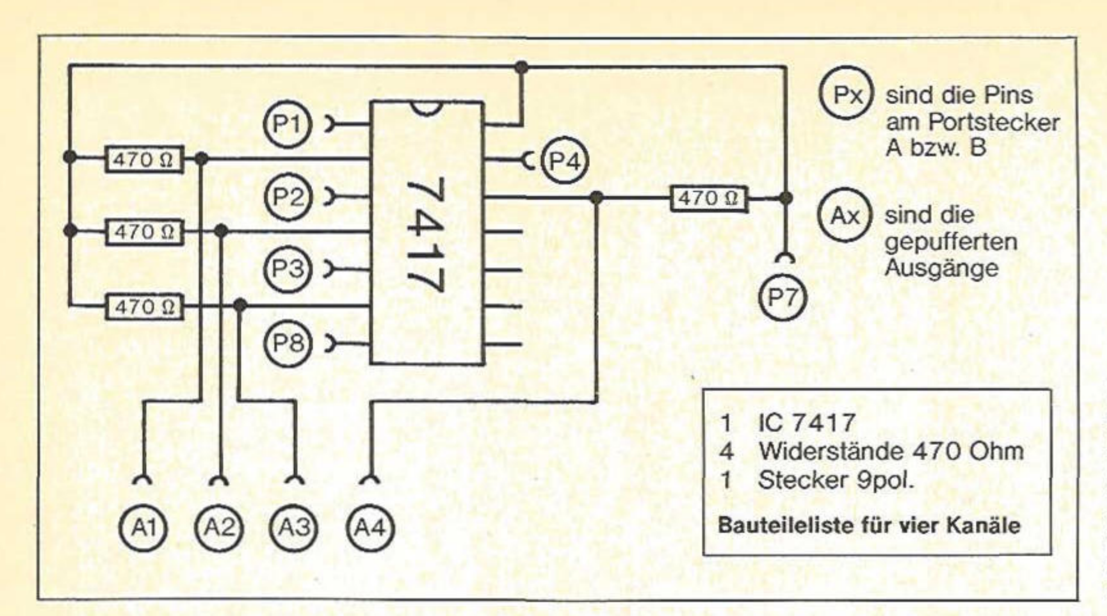


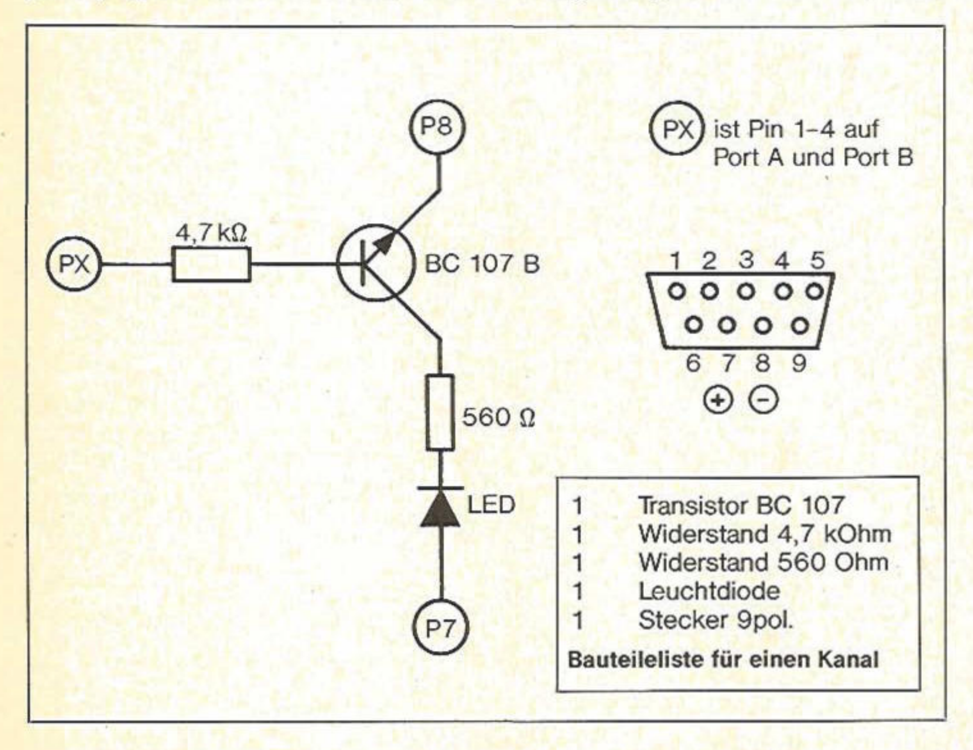
Bild 2. Schaltplan zum Ansteuern von vier Kanälen. Die abgedruckte Schaltung wird in zweifacher Ausführung benötigt, falls Sie alle acht Kanäle ansprechen möchten.

Die Programmierung der PIA geht folgendermaßen vor sich. Zuerst teilt man dem entsprechenden Datenrichtungsregister mit, daß die nächste Anweisung an den Port die Datenflußrichtung festlegen soll. Dies geschieht, indem man den Wert 48 in das Register POKEt. Der Wert für die gewünschte Flußrichtung ergibt sich dadurch, daß man für jedes Datenbit, das auf Ausgabe programmiert werden soll, eine logische Eins und für jede Datenleitung zur Eingabe eine logische Null eingibt. Will man also beispielsweise alle acht Leitungen zur Ausgabe benutzen, ergibt dies den Wert 255 (oder binär

Daten rein oder raus?

11111111). Für die Eingabe aller acht Leitungen ergäbe sich also der Wert Null (binär 00000000). Man kann also ohne weiteres auch beide Flußrichtungen gleichzeitig festlegen, indem man sich den entsprechenden Wert errechnet.

Der nächste Befehl bezieht sich nun wieder auf das Datenrichtungsregister. Man stellt den Normalzustand durch POKEn des Wertes 52 her. Nun ist das Port-Register nur noch für die logischen Zustände der Ein- und Ausgabe verantwortlich. Belegt man diese Speicherstelle also beispielsweise mit dem Wert Null, so würden alle Ausgänge den logischen Wert Null annehmen. Die Hardware, die man benötigt, um tatsächlich externe Vorgänge zu steuern oder zu erfassen, kann sich relativ einfach gestalten. Die Logikpegel der PIA entsprechen nämlich dem TTL-Standard. Dies bedeutet, daß die auszugebenden Informationen ohne Probleme durch recht preiswerte integrierte Bausteine weiterzuverarbeiten sind. Bei der Eingabe genügt es sogar im einfachsten Fall (wie es beispielsweise bei den Joysticks praktiziert wird), einen Eingang auf +5 Volt zu legen, um ein logisches Signal »Eins« zu erhalten.



▲ Bild 3. Gepufferter Ausgang für einen Kanal. Entsprechend werden acht Schaltungen benötigt, um alle Kanäle anzusprechen.

```
100 REM ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲     <OI>
110 REM ▲ZUSATZTASTATUR▲UEBER▲        <MW>
120 REM ▲DIE▲PIA▲ABFRAGEN▲▲▲▲▲▲       <AR>
130 REM ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲     <OO>
140 REM ▲von▲Wolfgang▲Czerny▲▲        <CC>
150 REM ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲     <OS>
160 GRAPHICS 0                        <SS>
200 X=PEEK(632)                       <AN>
210 IF X=0 THEN 200                   <WU>
220 ? "DIE▲GETIPPTE▲ZAHL▲LAUTET▲:";X-1 <VS>
230 GOTO 200                          <LX>
```

◀ Listing 1. Ein kurzes Beispielsprogramm zur Tastaturabfrage

Die Zusatztastatur

Als ein Beispiel, wie man externe Daten aufnehmen kann, wird hier eine Zusatztastatur beschrieben. Diese Tastatur besteht aus 12 Tasten, könnte aber auf bis zu 15 erweitert werden. Neben den Tasten besteht diese Schaltung im wesentlichen aus nur drei ICs. Einem 7400, er enthält vier NAND-Gatter und zwei ICs des Typs 74148. Dies sind sogenannte »8 zu 3 Prioritätsencoder« (der Schaltplan für die Zusatztastatur ist Bild 1 zu entnehmen). Der Vorteil dieser Schaltung liegt darin, daß ein einziger Joystickport ausreicht. Es werden nur vier Datenleitungen be-

nötigt, um die maximal 16 verschiedenen Zustände zu melden. Die Versorgungsspannung für die ICs wird dem Joystick-Port entnommen, so daß man ohne zusätzliches Netzteil auskommt. Listing 1 zeigt, wie man die Tastatur abfragen kann. Ist keine Taste gedrückt, erhält man den Wert Null, ansonsten einen Wert zwischen 1 und 11.

Auch die Datenausgabe ist hardwaremäßig einfach zu realisieren. Die Schaltung in Bild 2 zeigt, wie man mit nur zwei ICs, acht Widerständen und acht Leuchtdioden bereits die logischen Zustände aller Datenleitungen sichtbar machen kann. Da sich die Gesamtschaltung aus zwei identischen Teilen zusammensetzt, sieht man in Bild 2 nur den Aufbau für vier Kanäle. Die beiden ICs des Typs 7417 übernehmen hierbei die Aufgabe der Pufferung. Sie sind notwendig, um die PIA vor Überlastung zu schützen. An die Ausgänge der Pufferbausteine kann man nun wie in unserem Beispiel, Leuchtdioden, aber auch Optokoppler oder TTL-kompatible Kleinrelais anschließen.

Daten sichtbar gemacht

Bis zu diesem Punkt kommt man auch mit dem Schaltinterface mit der Stromversorgung der Joystickports aus. Es ist somit kein zusätzliches Netzteil erforderlich. Begnügt man sich mit der Steuerung von Leuchtdioden, so kann man auch die Schaltung in Bild 3 verwenden. Auch hier gilt: Die abgedruckte Schaltung in Bild 3 bezieht sich wieder auf nur einen Kanal.

Pro Datenleitung benötigt man zwei Widerstände, einen Transistor und eine Leuchtdiode. Auch hier wird ein Pufferungseffekt erreicht, da der Strom der Leuchtdiode nicht mehr über die PIA, sondern über den Transistor läuft.

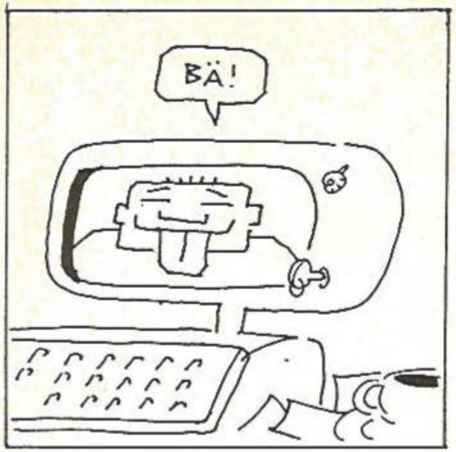
Das etwas umfangreichere Listing 2 zeigt, wie man diese Schaltungen ansteuern kann. Neben einem programmierbaren Lauflicht ist auch eine Zeitschaltuhr realisierbar, die eine der Datenleitungen zu einem vorgewählten Zeitpunkt ein- oder ausschaltet. Um das Lauflicht schnell genug zu machen, ist eine Maschinenroutine in das Listing aufgenommen worden. Sie verschiebt den Registerinhalt des Ports A jeweils um 1 Bit nach links oder rechts, abhängig von der Einsprungstelle. Alles weitere kann man dem Listing selbst problemlos entnehmen.

(Wolfgang Czerny/wb)

Listing 2. Mit diesem Programm kann die Schaltung von Bild 2 als Schaltuhr eingesetzt werden. Weiterhin ist noch eine Lauflicht-Funktion eingebaut. ▶

```
100 REM _____PIA__KONTROLLE_________          <SD>
110 REM ____________________________          <DI>
120 REM _54016_=_PORTA______________          <HM>
130 REM _54018_=_PORTACNTL__________          <UB>
140 REM ____________________________          <DO>
150 REM _POKE_54018,48_SETZT_PORT_A_          <WM>
160 REM ______AUF_DATENRICHTUNGS-___          <ZZ>
170 REM ______REGISTER_DDRA_________          <UW>
180 REM _POKE_54016,255_SETZT_ALLE__          <LQ>
190 REM ______8_BITS_AUF_AUSGABE____          <UD>
200 REM _POKE_54018,52_SETZT_DDRA___          <AJ>
210 REM ______ZURUECK_AUF_PORT_A____          <BA>
220 REM _POKE_54016,0_SETZT_ALLE____          <TW>
230 REM ______BITS_AUF_LOGISCH_NULL_          <ME>
240 REM ____________________________          <DP>
250 REM _von_Wolfgang_Czerny________          <RD>
260 REM ____________________________          <DT>
300 P1=54016:P2=54018:DIM FL(7),DAT$(40)
,EIN$(3)                                      <ZW>
310 GOSUB 9000:GOSUB 10000                    <VG>
490 REM ----------------------------          <ZN>
500 REM HAUPTMENUE                            <WN>
510 GRAPHICS 0:POKE 752,1                     <HA>
515 POKE 709,255:POKE 710,0                   <CH>
520 RESTORE 11000                             <AP>
530 FOR I=2 TO 16:READ DAT$                   <NI>
540 POSITION 8,I:? DAT$                       <AQ>
550 NEXT I                                    <GF>
700 OPEN #1,4,0,"K:":GET #1,K:CLOSE #1        <UF>
710 IF K<49 OR K>53 THEN 700                  <KK>
720 ON K-48 GOSUB 1000,2000,9000,5000,15
000                                           <TG>
800 GOTO 500                                  <NB>
999 REM ----------------------------          <AT>
1000 REM _LAUFLICHT_                          <GI>
1010 ? CHR$(125):POKE 752,1:RESTORE 1200
0                                             <SY>
1020 POSITION 14,2:? "LAUFLICHT"              <CT>
1030 POSITION 14,3:? "{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}"                                        <XI>
1040 FOR I=5 TO 11:READ DAT$                  <VY>
1050 POSITION 3,I:? DAT$                      <CE>
1060 NEXT I                                   <FH>
1110 READ DAT$:POSITION 2,15:? DAT$           <QO>
1120 POSITION 2,16:? "{SHIFT =}_";CHR$(2
7);CHR$(30);"_Cursur_links_{SHIFT =}_";C
HR$(27);CHR$(31);"_Cursor_rechts_{SHIFT
=}"                                           <LA>
1130 FOR I=17 TO 21:READ DAT$                 <GW>
1140 POSITION 2,I:? DAT$                      <BO>
1150 NEXT I                                   <FG>
1180 POSITION 33,10:? "C"                     <BW>
1190 POKE 764,255:F=0:CP=33                   <IP>
1195 FOR I=0 TO 7:FL(I)=0:NEXT I              <JT>
1200 TC=PEEK(764):SC=PEEK(53279)              <WK>
1210 IF TC=6 THEN POSITION CP,10:? "_":C
P=CP-4:F=F+1:GOTO 1300                        <HN>
1220 IF TC=7 THEN POSITION CP,10:? "_":C
P=CP+4:F=F-1:GOTO 1300                        <IO>
1230 IF TC=28 OR TC=33 THEN FL(F)=(33=TC
):GOTO 1400                                   <SQ>
1250 IF SC=6 THEN 1500                        <EN>
1260 IF SC=5 THEN 500                         <OQ>
1270 IF SC=3 THEN EDIT=1:GOTO 1000            <MW>
1280 GOTO 1200                                <OO>
1300 IF F<0 THEN F=7:CP=5                     <FG>
1310 IF F>7 THEN F=0:CP=33                    <AU>
1340 POSITION CP,10:? "C"                     <CQ>
1350 POKE 764,255:GOTO 1200                   <TL>
1400 IF FL(F)=0 THEN POSITION CP,7:? "_"      <PD>
1410 IF FL(F)=1 THEN POSITION CP,7:? "_*
"                                             <EV>
1420 POKE 764,255:GOTO 1200                   <TE>
1500 POSITION 2,23:? "L=LINKSROTATION_/_
R=RECHTSROTATION";                            <EN>
1510 OPEN #1,4,0,"K:":GET #1,K:CLOSE #1       <YY>
1520 IF K<>76 AND K<>82 THEN 1510             <DC>
1530 IF K=76 THEN SHIFT=1536                  <BA>
1540 IF K=82 THEN SHIFT=1555                  <AS>
1690 REM BINAER-DEZ UMWANDLUNG                <XD>
1700 WERT=0:FOR I=0 TO 7:X=FL(I):WERT=WE
RT+2^I*X:NEXT I:POKE P1,WERT                  <MZ>
1705 REM START LAUFLICHT                      <JA>
1710 SC=PEEK(53279)                           <XL>
1720 IF SC=3 THEN EDIT=1:GOTO 1000            <MR>
1730 IF SC=5 THEN GOTO 500                    <GZ>
1750 GOSUB 8000:X=USR(SHIFT)                  <TR>
1760 GOTO 1710                                <RN>
1999 REM ----------------------------         <EW>
2000 REM BINAERZAEHLER                        <WB>
2010 FOR Y=1 TO 10:FOR I=0 TO 255:POKE P
1,I:GOSUB 8000:NEXT I:NEXT Y                  <QY>
2020 RETURN                                   <PA>
2999 REM ----------------------------         <EX>
5000 REM SCHALTUHR                            <WY>
5010 ? CHR$(125)                              <GC>
5020 POSITION 2,7:? "WELCHEN_KANAL_SCHAL
TEN_(1-8)_";:INPUT KANAL                      <GA>
5030 IF KANAL<1 OR KANAL>8 THEN 5010          <YR>
5035 POSITION 2,9:? "WOLLEN_SIE_EIN-_ODE
R_AUSSCHALTEN"                                <AX>
5040 POSITION 2,11:? "EIN=1_AUS=0_";:INP
UT EIN                                        <WT>
5050 IF EIN<0 OR EIN>1 THEN 5040              <TO>
5090 IF EIN=0 THEN POKE P1,255                <AD>
5100 IF EIN=1 THEN POKE P1,0                  <HG>
5110 POSITION 2,13:? "GEBEN_SIE_DIE_MOME
NTANE_UHRZEIT_EIN"                            <UF>
5120 POSITION 2,14:? "(STD,MIN,SEC)_";:I
NPUT STD,MIN,SEC                              <KD>
5130 IF STD>23 OR STD<0 OR MIN>59 OR MIN
<0 OR SEC>59 OR SEC<0 THEN 5120               <TC>
5140 UP1=(STD*3600+MIN*60+SEC)*50             <EK>
5150 UP2=INT(UP1/256/256)                     <FC>
5155 UP1=UP1-UP2*256*256                      <DG>
5160 UP3=INT(UP1/256):UP1=UP1-UP3*256         <KL>
5165 POKE 18,UP2:POKE 19,UP3:POKE 20,UP1      <IJ>
5170 POSITION 2,16:? "GEBEN_SIE_DIE_SCHA
LTZEIT_EIN"                                   <SL>
5180 POSITION 2,17:? "(STD,MIN,SEC)_";:I
NPUT STD1,MIN1,SEC1                           <KX>
5190 IF STD1>23 OR STD1<0 OR MIN1>59 OR
MIN1<0 OR SEC1>59 OR SEC1<0 THEN 5180         <JB>
5200 ? CHR$(125)                              <GD>
5210 POSITION 11,4:? "SCHALTUHR_LAEUFT"       <JN>
5220 POSITION 11,5:? "----------------"       <GP>
5230 IF EIN=0 THEN EIN$="AUS"                 <FY>
5240 IF EIN=1 THEN EIN$="EIN"                 <SS>
5250 POSITION 1,8:? "KANAL_";KANAL;"_WIR
D_UM_":STD1:":":MIN1;":":SEC1;"_":EIN$;"
GESCHALTET"                                   <UF>
5300 U1=PEEK(18):U2=PEEK(19):U3=PEEK(20)      <DD>
5310 U1=INT((U1*256*256+U2*256+U3)/50)        <BM>
5320 U2=INT(U1/3600):U1=U1-U2*3600            <RO>
5330 U3=INT(U1/60):U1=U1-U3*60                <XG>
5340 U2=U2-24*(U2>=24)                        <HM>
5350 POSITION 10,12:? U2;":";U3;":";U1        <KY>
5360 POSITION 10,12:? U2;":";U3;":";U1        <LB>
5380 IF STD1<>U2 OR MIN1<>U3 OR SEC1<>U1
 THEN 5300                                    <MF>
5390 IF EIN=1 THEN POKE P1,2^(KANAL-1)        <DH>
5400 IF EIN=0 THEN POKE P1,0                  <GZ>
5900 RETURN                                   <PP>
7999 REM ----------------------------         <FC>
8000 REM VERZOEGERUNGSSCHLEIFE                <IK>
8010 FOR DELAY=1 TO 50:NEXT DELAY             <OF>
8020 RETURN                                   <PG>
8999 REM ----------------------------         <FD>
9000 REM LOESCHEN LEDS PORT 1/2               <OO>
9010 POKE P2,48                               <DI>
9020 POKE P1,255                              <GU>
9030 POKE P2,52                               <AQ>
9040 POKE P1,0                                <UT>
9050 RETURN                                   <PQ>
9999 REM ----------------------------         <FE>
10000 REM Maschinen File laden                <LY>
10010 RESTORE 10100                           <EU>
10020 FOR A=1536 TO 1579:READ D:POKE A,D
:NEXT A                                       <QX>
10030 REM LINKS=1536 / RECHTS=1555            <JL>
10090 RETURN                                  <UH>
10100 DATA_104,24,173,0,211,10,176,3,76,
39,6,141,43,6,169,1,76,35                     <IU>
10110 DATA_6,104,24,173,0,211,74,176,3,7
6,39,6,141,43,6,169,128,24                    <PI>
10120 DATA_109,43,6,141,0,211,96,0            <QD>
11000 REM DATAS_HAUPTMENUE                    <RI>
11010 DATA                                    <LH>
11020 DATA_HAUPTMENU                          <ZH>
11030 DATA                                    <LP>
11040 DATA                                    <LT>
11050 DATA_1_-_LAUFLICHT                      <JV>
11060 DATA                                    <MB>
11070 DATA_2_-_BINAER_LAUF                    <HE>
11080 DATA                                    <MJ>
11090 DATA_3_-_ALLE_LOESCHEN                  <MB>
11100 DATA                                    <LG>
11110 DATA_4_-_SCHALTUHR                      <ZY>
11120 DATA                                    <LO>
11130 DATA_5_-_ENDE                           <ZV>
11140 DATA                                    <LW>
11150 DATA                                    <MA>
12000 REM DATAS_LAUFLICHT-MENUE               <BV>
12010 DATA_{CTL Q}{CTL R}{CTL R}{CTL R}{
CTL W}{CTL R}{CTL R}{CTL R}{CTL W}{CTL R
}{CTL R}{CTL R}{CTL W}{CTL R}{CTL R}{CTL
 R}{CTL W}{CTL R}{CTL R}{CTL R}{CTL W}{C
TL R}{CTL R}{CTL R}{CTL W}{CTL R}{CTL R}
{CTL R}{CTL W}{CTL R}{CTL R}{CTL R}{CTL   <QH>
12020 DATA_{SHIFT =}___{SHIFT =}___{SHIF
T =}___{SHIFT =}___{SHIFT =}___{SHIFT =}
___{SHIFT =}___{SHIFT =}___{SHIFT =}      <ML>
12030 DATA_{SHIFT =}___{SHIFT =}___{SHIF
T =}___{SHIFT =}___{SHIFT =}___{SHIFT =}
___{SHIFT =}___{SHIFT =}___{SHIFT =}      <MP>
12040 DATA_{SHIFT =}___{SHIFT =}___{SHIF
T =}___{SHIFT =}___{SHIFT =}___{SHIFT =}
___{SHIFT =}___{SHIFT =}___{SHIFT =}      <MT>
12050 DATA_{CTL A}{CTL R}{CTL R}{CTL R}{
CTL S}{CTL R}{CTL R}{CTL R}{CTL S}{CTL R
}{CTL R}{CTL R}{CTL S}{CTL R}{CTL R}{CTL
 R}{CTL S}{CTL R}{CTL R}{CTL R}{CTL S}{C
TL R}{CTL R}{CTL R}{CTL S}{CTL R}{CTL R}
{CTL R}{CTL S}{CTL R}{CTL R}{CTL R}{CTL   <DR>
12060 DATA_{SHIFT =}___{SHIFT =}___{SHIF
T =}___{SHIFT =}___{SHIFT =}___{SHIFT =}
___{SHIFT =}___{SHIFT =}___{SHIFT =}      <NB>
12070 DATA_{CTL Z}{CTL R}{CTL R}{CTL R}{
CTL X}{CTL R}{CTL R}{CTL R}{CTL X}{CTL R
}{CTL R}{CTL R}{CTL X}{CTL R}{CTL R}{CTL
 R}{CTL X}{CTL R}{CTL R}{CTL R}{CTL X}{C
TL R}{CTL R}{CTL R}{CTL X}{CTL R}{CTL R}
{CTL R}{CTL X}{CTL R}{CTL R}{CTL R}{CTL   <ZN>
12080 DATA_{CTL Q}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL W}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL E}                   <BH>
12100 DATA_{CTL A}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL S}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL D}                   <MN>
12110 DATA_{SHIFT =}_Space__=__SET__{SHI
FT =}_Esc_=_UNSET_____{SHIFT =}           <VQ>
12120 DATA_{CTL A}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL W}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL X}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL W}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL D}                   <DE>
12130 DATA_{SHIFT =}START=RUN{SHIFT =}SE
LECT=Menu{SHIFT =}OPTION=Edit_{SHIFT =}   <AU>
12140 DATA_{CTL Z}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL X}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL X}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL C}                   <IR>
15000 END                                     <JY>
```







Roboter im Griff

Mit einer einfachen und preiswerten Schaltung wird der Atari zur Steuerungseinheit für Elektromotoren. Damit läßt sich ein Robotarm dirigieren.

Industrieroboter übernehmen heute in vielen Fabriken die schmutzigen und schweren Arbeiten. Ein Atari wäre bei diesen komplizierten Produktionsabläufen natürlich überfordert. Doch für die Steuerung von kleinen Modellrobotern ist er hervorragend geeignet. Einen einfachen Robotarm zu bauen, ist gar nicht so schwierig. Die Konstruktionsbaukästen von Märklin, Trix oder von Fischertechnik, sind eine gute Grundlage. Zusätzlich benötigt man natürlich noch die entsprechenden Motoren und Getriebe. Einige Punkte sind bei der Planung zu beachten:

1. Es sollten nur Gleichspannungsmotoren eingesetzt werden (6-12 V)
2. Alle Motore brauchen ein Getriebe, möglichst mit Schneckenrad-Antrieb
3. Stabiler, mechanischer Aufbau mit großer Grundplatte
4. Beim Verlegen der Kabel Bewegungsachsen beachten
5. Geeignete Stromquellen benutzen (Netzteile)

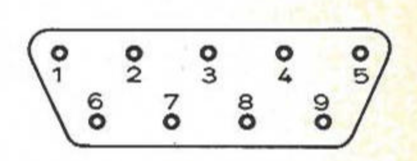
Entwerfen wir ein Konzept:

☐ Anzahl der Freiheitsgrade: Die Anzahl ist abhängig von der Funktion. Drei Freiheitsgrade sind in der Regel ausreichend, das heißt, drei bewegliche Achsen und drei Motore. Sie sind nötig zum Drehen des Grundkreises, heben und senken der senkrechten Achse, aus- und einfahren der waagerechten Achse.

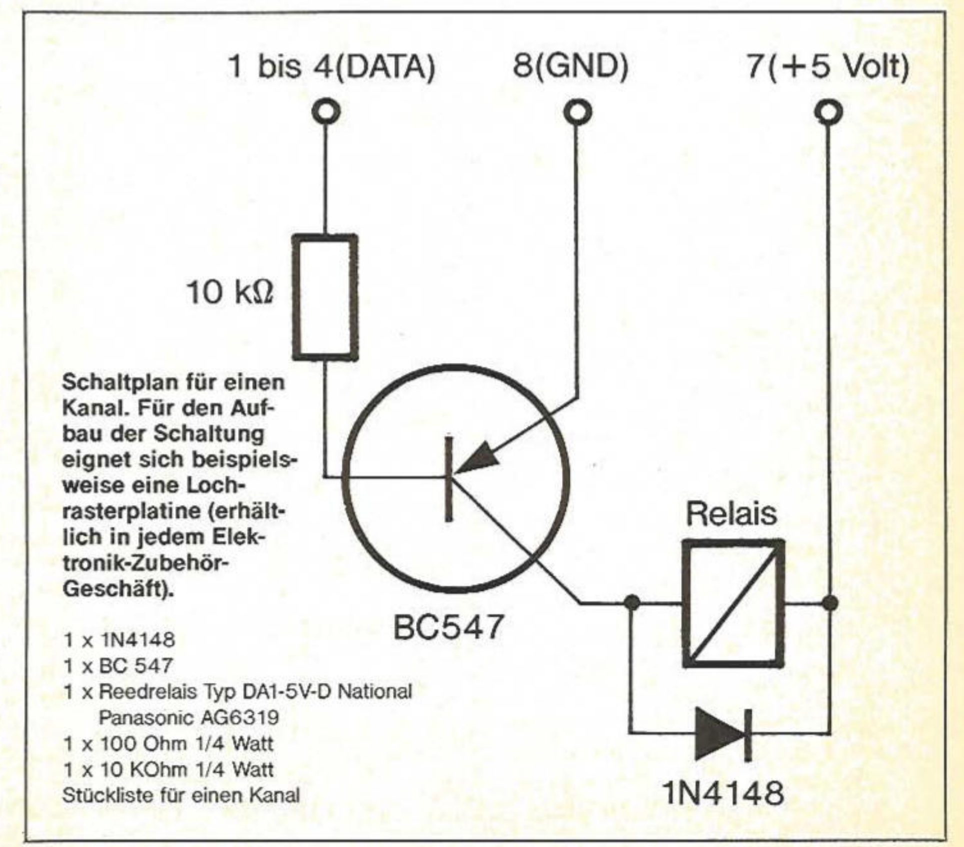
☐ Größe des Verfahrweges: Der Verfahrweg ist individuell zu gestalten, er ist verantwortlich für den Aktionsradius des Armes.

☐ Positionierung eines Robotarmes: Um den Geldbeutel zu schonen, verwenden wir keine Schrittmotoren. Besitzer eines Atari 400/800 können

Pins	Beschreibung
1 bis 4	DATA
7	+5 Volt maximal 40 mA
8	GND (Masse)



Die Anschlußbelegung eines Joystickports



Schaltplan für einen Kanal. Für den Aufbau der Schaltung eignet sich beispielsweise eine Lochrasterplatine (erhältlich in jedem Elektronik-Zubehör-Geschäft).

1 x 1N4148
1 x BC 547
1 x Reedrelais Typ DA1-5V-D National Panasonic AG6319
1 x 100 Ohm 1/4 Watt
1 x 10 KOhm 1/4 Watt
Stückliste für einen Kanal

sich freuen, weil hier Port 3 und 4 noch frei sind. XL-Besitzer dagegen müssen sich mit der Steuerung über den Timer zufriedengeben.

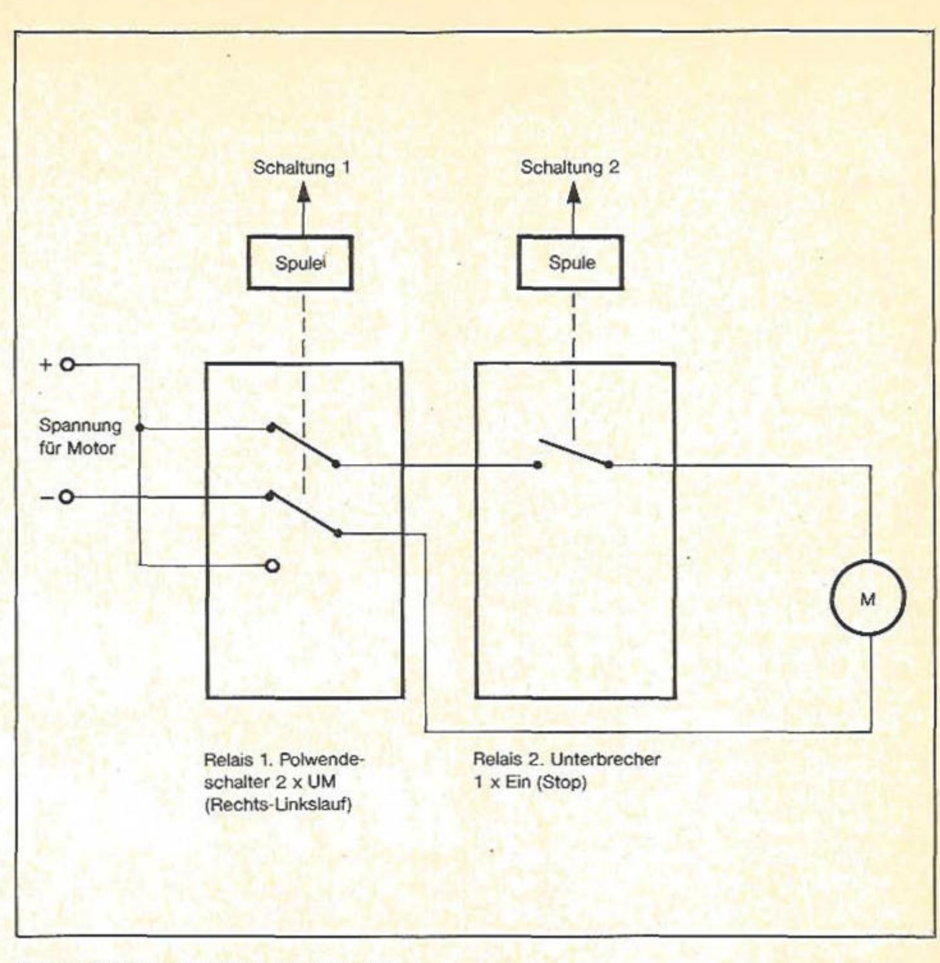
Der Atari-Computer ist natürlich nicht zum Steuern eines Robotarms ausgelegt. Wir müssen deshalb eine kleine Schaltung aufbauen (siehe Schaltplan). Damit sind wir in der Lage, den Arm von Basic aus zu steuern. Die Schaltung wird über Port 1 und 2 angeschlossen. Die Ports werden mit »POKE 54018,1« auf Datenausgabe umgeschaltet. Aber Vorsicht, nach Ausführung dieses POKE-Befehls darf kein Joystick mehr angeschlossen sein. Mit »POKE 54016,0-255« werden die Datenkanäle geschaltet. Jetzt steht uns eine Parallel-Schnittstelle mit 8 Bit zur Verfügung. Liegt ein Datensignal an, schaltet der Transistor die computereigene 5-Volt-Spannung auf das Reedrelais, welches dann anzieht. Durch sinnvollen Einsatz der POKEs kann man allen Motoren gleichzeitig ohne Interrupt-Programmierung ansteuern. Beim Atari 400 oder 800 kann man zusätzlich noch Potentiometer an alle Achsen anbauen (Wert 2MOhm) und über Port 3 und 4 den Wert erfassen. Damit ist es möglich den Arm relativ genau zu positionieren.

Alle Atari-XL-Benutzer können nur über die Zeit positionieren und haben den Nachteil ungenauer Steuerung. Wird nämlich der Motor belastet (zum Beispiel durch ein Gewicht), läuft er langsamer und erreicht die gewünschte Position nicht mehr. Hardwarespezialisten können versuchen über die neue Schnittstelle zum Erfolg zu gelangen.

Das Basic-Programm ist für zwei Motoren ausgelegt und arbeitet mit einem Joystick in Port 4. Mit dem Joystick sind alle Bewegungen zu steuern. Pfiffige können das Programm so umschreiben, das sogar ein »Teach in« möglich wird. Das heißt, steuern des Armes zur gewünschten Position mit dem Joystick, übernehmen des Wertes in das Programm und danach automatischer Ablauf des Vorgangs. Als Greifer leistet ein umgebautes Relais gute Dienste. Damit kann man metallische Gegenstände greifen und transportieren. Ein preiswerter Elektromagnet funktioniert natürlich auch.

Zum Schluß noch einige Tips: Die Joystickports sollten pro Kanal nicht höher als mit 10 mA belastet werden. Die Spule der Relais muß also die Werte 5 V/500 Ohm haben. Die Positionierungspotis sollten untersetzt werden, damit der gewünschte Punkt nicht überfahren wird. Auch alle Getriebe sollten über Schnecken und Schneckenräder untersetzt sein, damit eine Bremse überflüssig wird.

(Jürgen Warschinsky/wb)



Schematischer Aufbau der Schaltung

```
4 PRINT CHR$(125)
5 POKE 54018,1:X=54016:POSITION 10,8:? "
BEWEGUNG:"
10 A=STICK(3)
20 IF A=15 THEN POKE X,245:POSITION 10,1
0:? "    STOP    "
30 IF A=14 THEN POKE X,249:POSITION 10,1
0:? "    UP      "
40 IF A=13 THEN POKE X,253:POSITION 10,1
0:? "    DOWN    "
50 IF A=11 THEN POKE X,247:POSITION 10,1
0:? "    LEFT    "
60 IF A=7 THEN POKE X,246:POSITION 10,10
:? "    RIGHT   "
70 IF A=9 THEN POKE X,255:POSITION 10,10
:? "DOWN & LEFT "
80 IF A=5 THEN POKE X,254:POSITION 10,10
:? "DOWN & RIGHT"
90 IF A=10 THEN POKE X,251:POSITION 10,1
0:? "  UP & LEFT "
100 IF A=6 THEN POKE X,250:POSITION 10,1
0:? " UP & RIGHT "
110 GOTO 10
```

Listing zur Ansteuerung des Selbstbau-Schaltinterfaces



Brandneue Bücher rund um den ATARI ST

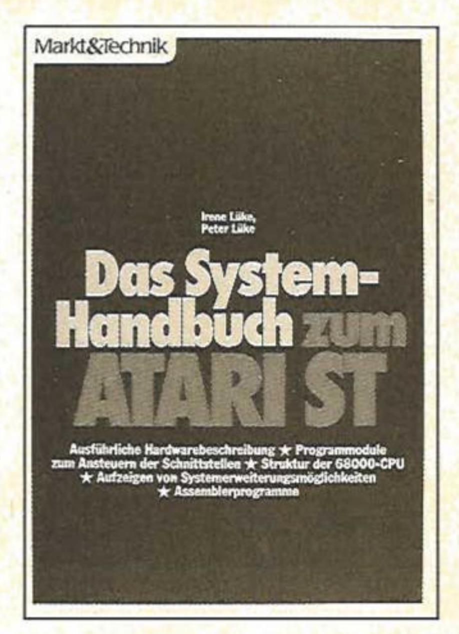


I. Lüke/P. Lüke
Der ATARI 520 ST
2. überarbeitete und erweiterte Auflage 1986, 198 Seiten

Dieses Buch enthält alle Informationen, die für Interessierte und für alle stolzen Besitzer eines gerade erworbenen ATARI 520/260 ST wichtig sind. Die jetzt vorliegende überarbeitete und erweiterte Auflage trägt den neuesten Entwicklungen bei Atari Rechnung. Unter anderem wurden das inzwischen deutschsprachige Betriebssystem und einige geänderte Systemausstattungsmerkmale berücksichtigt. Das Buch ist somit nicht nur eine Rechnerbeschreibung mit hohem Informationswert, es leistet auch als Nachschlagewerk wertvolle Dienste.
Best.-Nr. MT 90229
ISBN 3-89090-229-4
DM 49,–/sFr. 45,10/öS 382,20

I. Lüke/P. Lüke
Das Systemhandbuch zum ATARI ST
2. Quartal 1986, ca. 300 Seiten

Zwei Themen bilden die Schwerpunkte des vorliegenden Buches:
Die Struktur der 68000-CPU und der ATARI 520/260 ST.
Die ausführliche Beschreibung der Architektur der 68000-Familie (68000, 68008, 68010, 68020) und ihrem Befehlssatz wird ergänzt durch einen Nachschlageteil mit zwei- bis dreizeiligen Beispielsequenzen. Auf dieser theoretischen Basis wird die Programmierumgebung des ATARI 520/260ST anhand vieler Beispielprogramme dargestellt. Die Entwicklung dieser Programme liefert dem Leser gleichzeitig eine Bibliothek mit Routinen zur Ansteuerung des Bildschirmteils, der Tonerzeugungsschaltung und der Schnittstellen (MIDI, V24, Tastatur, Maus). Besondere Aufmerksamkeit wird der Einbindung von Maschinensprachmodulen in das Betriebssystem und in höhere Programmiersprachen (z. B. BASIC und C) gewidmet. Die Besprechung eines 68000-Assemblers und einige gerätespezifische Maschinensprachmodule runden das Buch ab.
Best.-Nr. MT 90216
ISBN 3-89090-216-2
DM 52,–/sFr. 47,80/öS 405,60





A. Steiner/G. Steiner
GEM für den ATARI 520 ST
2. überarbeitete und erweiterte Auflage 1986, 334 Seiten

Die Benutzeroberfläche des neuen ATARI ST – GEM genannt – erhebt den Anspruch, die Bedienung des Computers zum Kinderspiel zu machen. Dennoch: Wenn Sie die bisher übliche kommandoorientierte Umgangsweise mit Ihrem Computer pflegten, so werden Sie eine Einführung in die Bedienung von Maus, Bildsymbolen und Fenster, wie sie dieses Buch liefert, zu schätzen wissen. Besonders interessant für den erfahrenen Anwender sind die Kapitel über den internen Aufbau von GEM mit seinen Pull-Down-Menüs, Fenstern und Symbolen.
Best.-Nr. MT 90230
ISBN 3-89090-230-8
DM 52,–/sFr. 47,80/öS 405,60



J. Purdum/T. Leslie
Die C-Programmbibliothek
1. Quartal 1986, ca. 320 Seiten

Dieses Buch erspart dem C-Programmierer Stunden mühseliger Kleinarbeit und hilft, effizientere Programme zu schreiben. Es ist in zwei Teile gegliedert. Der erste Teil zeigt, wie man zu universellen Bibliotheksfunktionen kommt und gibt Tips, wie C noch wirkungsvoller eingesetzt werden kann. Der zweite Teil enthält eine Reihe ausführlich erklärter C-Funktionen als wertvolle Ergänzung Ihrer Programmbibliothek. Dazu gehören unter anderem ein Terminalinstallationsprogramm, mehrere Sortier-Algorithmen und ein Satz ISAM-Funktionen.
Best.-Nr. MT 90133
ISBN 3-89090-133-6
DM 69,–/sFr. 63,50/öS 538,20

W. Fastenrath
ATARI ST BASIC-Handbuch
1. Quartal 1986, ca. 250 Seiten

Suchen Sie eine Anleitung zur intensiven Ausnutzung der Fähigkeiten des ATARI 520/260 ST? Dann ist dieses Buch genau das richtige für Sie! Sie erfahren alles über das BASIC-System des ATARI ST. Jeder Befehl wird mit Programmbeispielen ausführlich erläutert. Den Schwerpunkt bildet eine Anleitung zur BASIC-Programmierung des ATARI ST sowie zur Programmierung von GEM-Funktionen.
Best.-Nr. MT 90205, ISBN 3-89090-205-7, DM 52,–/sFr. 47,80/öS 405,60

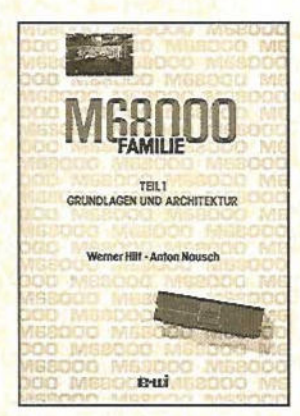
R. Aumiller
ATARI ST LOGO
1. Quartal 1986, ca. 250 Seiten

Dieses Buch bietet eine gründliche Einführung in die Programmmiersprache LOGO und ihre Anwendung auf dem ATRI 520/260 ST. Schon nach kurzer Zeit ist der Anfänger in der Lage, eigene LOGO-Programme zu schreiben. Praktische Anwendungsmöglichkeiten wie z. B. die Datenverwaltung sind auch für den fortgeschrittenen Programmierer von Interesse. Ein eigenes Kapitel ist dem Bereich der künstlichen Intelligenz gewidmet.
Best.-Nr. MT 90223, ISBN 3-89090-223-5, DM 49,–/sFr. 45,10/öS 405,60

P. Rosenbeck
C-Programmierung unter TOS/ATARI ST
1. Quartal 1986, ca. 300 Seiten

Die Programmiersprache C hat sich bei professionellen Programmierern zu einem Renner entwickelt. Sie ermöglicht es, sehr nahe an der Maschine zu arbeiten und doch strukturiert zu programmieren. Dieses Buch bietet eine Einführung in die Programmierung C speziell für den ATARI ST. Am Beispiel eines Diskettenmonitors wird die Systemprogrammierung gründlich und umfassend erläutert. Außerdem erfahren Sie alles über den Einsatz von BIOS-Routinen und über das Software-Engineering.
Best.-Nr. MT 90226, ISBN 3-89090-226-X, DM 52,–/sFr. 47,80/öS 405,60

In Vorbereitung:
C-Programmierung unter GEM/ATARI ST
2. Quartal 1986, ca. 300 Seiten
Best.-Nr. MT 90203, ISBN 3-89090-203-0, DM 58,–/sFr. 53,40/öS 452,40



W. Hilf/A. Nausch
M68000-Familie: Teil 1
1984, 568 Seiten

Informative Einführung in die Geschichte und die Entwicklungsphilosophie einer detaillierten Darstellung der Hardware sowie ausführliche Erläuterung der komfortablen Adressierungsarten.
Best.-Nr. PW 705
ISBN 3-921803-16-0
DM 79,–/sFr. 72,80/öS 616,20

M68000-Familie: Teil 2
1985, 400 Seiten

Teil II des umfassenden Lehr- und Nachschlagewerks zum M68000 beschäftigt sich mit Anwendungen und weiteren Mitgliedern der M68000-Familie.
Best.-Nr. PW 713
ISBN 3-921803-30-6
DM 69,–/sFr. 63,50/öS 538,20

Markt & Technik-Fachbücher erhalten Sie bei Ihrem Buchhändler

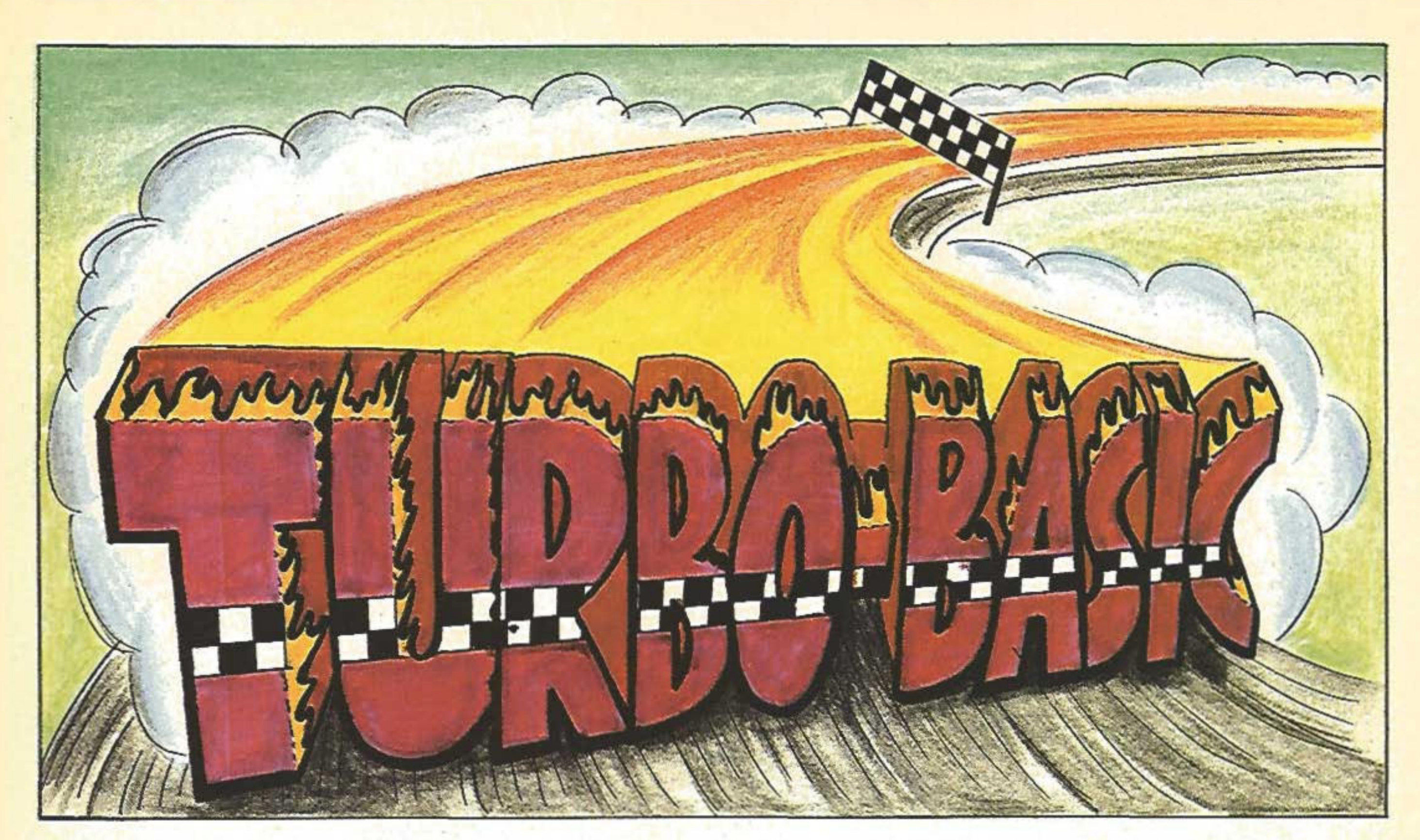
Bestellkarten bitte an Ihren Buchhändler oder an einen unserer Depot-Händler. Adressenverzeichnis am Ende des Heftes.

Bestellungen im Ausland bitte an untenstehende Adressen.
Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, ☎ 042/41 56 56
Österreich: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, ☎ 0222/67 75 26

Irrtümer und Änderungen vorbehalten.



Hans-Pinsel-Straße 2, 8013 Haar bei München



Der große Turbo-Basic-Teil

Turbo-Basic XL ist extrem schnell, komfortabel und erlaubt Programmstrukturen wie Pascal. Der Turbo-Basic-Compiler, der sich nun zur Turbo-Serie hinzugesellt hat, steigert das Programmiervergnügen und die Geschwindigkeit noch einmal.

Turbo-Basic XL werden manche Leser bereits kennen (der Interpreter wurde in Happy-Computer, Ausgabe 12/85 abgedruckt). Eventuell haben Sie auch schon erste Erfahrungen mit diesem leistungsfähigen Interpreter gesammelt. Dann werden Sie bestimmt nicht mehr allzuhäufig mit dem normalen Atari-Basic arbeiten und lieber auf das Turbo-Basic XL zurückgreifen. Aus diesem Grund veröffentlichen wir den Interpreter noch einmal in diesem Sonderheft. Wir sind nämlich der Meinung, daß sich Turbo-Basic XL unter möglichst vielen Atari-Fans verbreiten sollte.

Wenn Sie den Interpreter allerdings noch nicht kennen, schenken Sie den nächsten Seiten Ihre Aufmerksamkeit. Dort finden Sie eine ausführliche Erklärung zu allen Turbo-Basic-XL-spezifischen Befehlen. Hinzu kommen einige neue Fehlermeldungen und die Möglichkeit, verschiedene Disketten-Befehle direkt von Basic auszuführen. Lassen Sie sich aber nicht von dem umfangreichen Listing abschrecken. Denn haben Sie das Programm erst einmal abgetippt, werden Sie es nicht bereuen. Dessen sind wir ganz sicher.

Gleich im Anschluß an den Turbo-Basic-XL-Interpreter finden Sie einen weiteren Leckerbissen: Den Turbo-Basic-Compiler, ebenfalls zum Abtippen. Einen superschnellen Compiler, der selbst in Atari-Basic geschriebene Programme verarbeitet. Das Compilat zählt mit zu den schnellsten. Überzeugen Sie sich selbst von den Ergebnissen. Auf der Seite 32 sind die erzielten Rechenzeiten aufgelistet. Unser Compiler hat selbst kommerzielle Compiler um Längen geschlagen.

Damit sich die beiden »Wunderkinder« auch gleich bewähren können, sind einige Turbo-Basic-XL-Programme hier abgedruckt. Beachten Sie aber bitte: Diese Beispielprogramme laufen nicht unter Atari-Basic! Es kommen nämlich spezielle Turbo-Basic XL-Befehle zur Anwendung, die Atari-Basic nicht verarbeiten kann. So ist beispielsweise im Programm »Grafikspielereien« der »MOVE«-Befehl eingebaut, der ein sehr schnelles Verlagern des Bilschirmspeichers bewirkt. So wird ein Grafikbild mal auf den Kopf gestellt oder über die Mittelachse hinweg gespiegelt. Aber lassen Sie sich überraschen! Das Programm bietet viele wirkungsvolle Effekte, die sich ansonsten in Atari-Basic nur durch spezielle Maschinensprach-Routinen realisieren lassen.

Oder werfen Sie einen Blick auf das Programm »Apfelmännchen«. Dieses Programm erzeugt die tollsten Grafiken. Ein farbiges Beispiel finden Sie im Inhaltsverzeichnis am Anfang des Heftes. Hier kommt ein weiterer Vorzug von Turbo-Basic XL voll zur Geltung, nämlich der, strukturiert zu programmieren. Es wurde vor allem der Befehl »EXEC Prozedur« eingesetzt. So lassen sich selbst Prozeduren unter Turbo-Basic XL definieren. Beim genaueren Studium des »Apfelmännchen«-Programms wird Ihnen sicherlich auffallen, daß in dem Programm kein einziger GOTO- oder GOSUB-Befehl auftaucht. Dies ist eine der wichtigsten Voraussetzungen der strukturierten Programmierung.

Es spricht also einiges zugunsten unseres Interpreters und Compilers. Um noch mehr Atari-Besitzer in den Genuß des Turbo-Basic XL-Interpreters zu bringen, ist eine Umsetzung der beiden Programme auch für den Atari 260 ST/520 ST+ in Vorbereitung. Natürlich ist in dieser Version auch die Befehls-Kompatibilität zwischen dem 800XL und den ST-Computern gewährleistet. Allerdings benötigen die Programme noch einige Zeit bis zur Vollendung. Wir halten Sie aber auf dem laufenden.

(wb)

So funktioniert der Turbo-Basic-Compiler

Soll ein Basic-Programm schneller werden, dann greift man einfach zum Basic-Compiler. Aber wie funktioniert ein solches Beschleunigungswunder?

Manche Anwendungen erfordern mehr Tempo, als reine Basic-Programme zu bieten haben. Dann schreibt man zeitkritische Routinen entweder in Maschinensprache oder setzt einen Compiler ein, der ein Basic-Programm in Maschinensprache umwandelt. Letzteres ist natürlich viel einfacher und bequemer, da es lediglich den Umgang mit dem Compiler vorraussetzt. Außerdem benötigt die Compilierung nicht viel Zeit. Das Compilat kann übrigens nicht mehr von Basic aus bearbeitet werden. Wenden wir uns nun der Funktionsweise des Turbo-Basic-Compilers zu.

Wenn Sie ein Basic-Programm eintippen, zum Beispiel die Programmzeile »100 ?3*4«, faßt der Computer diese Zeile zunächst nur als eine Folge von Zeichen im ATASCII-Format auf. Unter dem Begriff ATASCII versteht man die spezielle Atari-Version des ASCII-Zeichensatzes (ASCII ist die Abkürzung für: American Standard Code for Information Interchange). Der Interpreter wandelt den eingegebenen Text in Token um. Gleichzeitig überprüft er noch die Syntax, also ob die Befehlsfolge auch zulässig ist. Diese Aufgabe muß meistens auch ein Compiler übernehmen. Bild 1 zeigt, wie die Programmzeile »100 ?3*4« in Token übersetzt aussieht.

Die Programmzeile wird dann anhand ihrer Nummer einsortiert. Durch diese Umwandlung in Token und gleichzeitiger Syntaxprüfung wird die Aufgabe des Compilers etwas einfacher. Speichert man ein Basic-Programm mit dem SAVE-Befehl, werden auch die Token gespeichert. Mit LIST hingegen liegt es in der Textform vor. Der Turbo-Basic-Compiler kann nur die »SAVE-Ausführung« übersetzen. Ein Transfer der Text-Form wäre prinzipiell auch möglich, allerdings würde der Compiler hierfür mehr Zeit benötigen. Schließlich muß er die Befehlsnamen kennen und die Variablennamen speichern. Dies würde weiterhin die maximale Länge der zu compilierenden Programme vermindern. Oder es müßten Zwischenfiles auf Diskette angelegt werden, was bei den relativ langsamen Atari-Diskettenlaufwerken sehr zeitaufwendig ist.

Keine Variable ohne Nummer

Bevor Sie ein Basic- oder Turbo-Basic XL-Programm compilieren, sollten Sie sicherstellen, daß es fehlerfrei läuft. Zwar ist dann immer noch nicht gewährleistet, daß das Programm anschließend auf Anhieb ordnungsgemäß compiliert und ausgeführt werden kann. Zumindest entfallen aber Syntax-Fehler, die zur Folge hätten, daß Sie Ihr Basic-Programm korrigieren müßten, um es anschließend nochmals zu compilieren. Auch muß die Struktur Ihrer Basic-Programme stimmen. Dazu bietet sich der LIST-Befehl unter Turbo-Basic XL an, der bekanntlich FOR-NEXT-Schleifen und IF-ELSE-ENDIF-Abfragen einrückt.

Atari-Basic und Turbo-Basic XL-SAVE-Files bestehen aus einem Vorspann mit den Variablennamen und einigen anderen Informationen für den Interpreter. Diese werden deshalb vor dem Einlesen entweder überlesen (wie die Variablennamen) oder, in einer angepaßten Form, gespeichert. Dies betrifft in erster Linie die Variablentypen. Der Variablentyp kann aber nicht nur aus der Variablennummer, die im Programm gespeichert ist, definiert werden. Das Dollarzeichen ($) beispielsweise, welches die Stringvariablen kennzeichnet, wird vom Interpreter nicht im eigentlichen Programm gespeichert.

Bei langen Programmen mit vielen Variablen kann man diese Phase der Compilation deutlich erkennen. In der Info-Zeile (der dritten Bildzeile des Turbo-Basic-Compilers) steht dann nur das Wort Zeile ohne Zeilennummer. Sobald die erste Zeile übersetzt wird, erscheint auch die Anzeige der entsprechenden Zeilennummer.

Jetzt beginnt die eigentliche Compilierung. Sie setzt sich aus zwei Durchgängen zusammen. Im ersten Durchgang (Paß 1) wird das Programm Zeile für Zeile und Befehl für Befehl eingelesen und in Maschinencode übersetzt. Im zweiten Durchgang (Paß 2) werden dann noch die offenen Sprungadressen eingesetzt. Dazu werden im Paß 1 GOTO-Sprünge im Code durch einen ungültigen Maschinenbefehl mit angehängter Zeilennummer gekennzeichnet. Im zweiten Durchgang schließlich ersetzen dann 6502-JMP-Befehle (Code $4C) die ungültigen Maschinenbefehle.

Ähnliches gilt für Zahlen und Textkonstanten. Sie werden in einem Block hinter dem eigentlichen Basic-Programm gespeichert. Die endgültigen Adressen stehen dann erst nach der Übersetzung fest. Im Anschluß an den zweiten Compilier-Durchgang folgt dann noch eine Kontrolle, ob offene Schleifen vorliegen. Also ob ein NEXT zu einem FOR fehlt oder ein WEND zu einem WHILE etc. Trifft dies zu, erscheint eine Meldung auf dem Bildschirm. Dann folgt noch die Überprüfung nach fehlenden ENDIFs. Durch diese Aufteilung der Fehlerprüfung (die meisten Fehler werden bereits im ersten Durchgang festgestellt) werden die fehlerhaften Zeilennummern nicht komplett sortiert ausgegeben, sondern gegebenenfalls in vier einzelnen Gruppen.

Nach der Umwandlung in Token sieht die Programmzeile »100 ? 3*4« folgendermaßen aus:

0A 00 15 15 28 0E 40 03 00 00 00 00 24 0E 40 04 00 00 00 00 16

Dabei bedeuten:

0A 00	= Zeilennummer (100)
15	= Zeilenlänge
15	= Abstand vom Zeilenanfang zum Ende der ersten Programmzeile
28	= Token für »?«
0E 40 03 00 00 00 00	= Zahl 3
24	= Token für »*«
0E 40 04 00 00 00 00	= Zahl 4
16	= Token für Zeilenende

Bild 1. Eine Basic-Programmzeile, umgesetzt in Token

Die meiste Arbeit wird im ersten Durchgang geleistet, der Rest dauert nur wenige Sekunden. Das Einlesen des Programms erfolgt Zeile für Zeile. Zuerst zur Zeilennummer: Wenn die Zeilennummer größer als 32767 ist, dann ist Paß 1 beendet (der Interpreter verwendet die Zeilennummer 32768 übrigens als Endekennzeichen). Dann schließt der zweite Durchgang an. Ansonsten wird die Zeilennummer angezeigt und die Zeile Befehl für Befehl eingelesen und übersetzt.

Sollte der Compiler auf einen LIST- oder ENTER-Befehl stoßen, wird einfach eine Fehlermeldung mit Zeilennummer ausgegeben, und der nächste Befehl kommt an die Reihe.

Keinerlei Probleme bereiten REM- oder »--«-Zeilen. Diese Befehle werden einfach überlesen und ignoriert, da sie den Programmablauf nicht beeinflussen.

Befehle ohne Parameter (wie END, POP oder CLR) werden einfach zu einem entsprechenden Unterprogrammaufruf compiliert.

Bei einem DATA-Befehl wird der Text mit Zeilennummer und Länge in einer Tabelle im Anschluß an das Programm eingetragen. Sie nimmt noch folgende Informationen auf:

- eine weitere Tabelle mit konstanten Zahlen
- konstanten Strings
- die Anfangsadressen aller Programmzeilen
- den während der Compilation benötigten Stapelspeicher (der 6502-Stapel ist viel zu klein)

Diese Informationen werden gegebenenfalls auch im Speicher verschoben, wenn das zu compilierende Programm oder eine der Tabellen anwächst. Deshalb nimmt auch die Compilierungsgeschwindigkeit bei langen Programmen gegen Ende ab; vor allem wenn viele DATAs am Anfang des Programms stehen. Diese Verlangsamung betrifft lediglich die eigentliche Compilierung und nicht den Programmablauf.

Alle Befehle, die Parameter erfordern, sind wesentlich schwieriger zu übersetzen. Bild 2 zeigt nur einen Befehl mit einem einzigen Parameter.

Die Umwandlung des Befehls von Schritt 1 nach Schritt 2 hat der Interpreter schon erledigt. Der Compiler liest das Befehlstoken $03 für COLOR und verzweigt in die entsprechende Compilationsroutine. Diese ruft wiederum die Routine für die Übersetzung eines Ausdrucks auf, die eine Integerzahl zurückliefern soll. Dann wird nur noch »STA $C8« angehängt, und damit ist die Compilierung des Befehls schon erledigt. Beim Befehl GRAPHICS würde »JSR @Graphics« angehängt werden. Der Klammeraffe (@) kennzeichnet übrigens eine Routine in der Runtime Bibliothek.

Ein großer Fortschritt wurde bis jetzt noch nicht verzeichnet. Zwar sind die Befehle jetzt prinzipiell compiliert, das ändert aber leider nichts daran, daß mit dem entsprechenden Parameter, wie in unserem Beispiel die 1, noch nichts passiert ist.

Compilieren Zeile für Zeile

Die dafür benötigte Routine gliedert sich in mehrere Abschnitte:

- P-Code-Erzeugung
- P-Code-Optimierung
- Maschinencode-Erzeugung

P-Code (Pseudo-Code) ähnelt der Maschinensprache eines hypothetischen (gedachten) Prozessors. Manche Compiler erzeugen nur einen P-Code, der dann von einem kleinen Interpreter ausgeführt wird. Die Abarbeitungsgeschwindigkeit eines solchen Codes ist zwar schneller als ein herkömmliches Basic-Programm, allerdings von der Geschwindigkeit einer »richtigen« Maschinensprache noch weit entfernt. Aber platzsparender ist P-Code im Vergleich zu voll compiliertem Basic. Echte Maschinensprache ist jedoch von keinem Compiler zu übertreffen.

Bild 3 zeigt ein komplizierteres Beispiel. »03« entspricht wieder dem Token für COLOR, »46« steht für PEEK, »3A« für die offene Klammer, »2C« für die schließende Klammer, »35« steht für die Addition (+), »16« bedeutet Zeilenende oder auch einen Doppelpunkt, der Basic-Befehle in einer Zeile trennt. Der Wert »0E« kennzeichnet schließlich noch eine Zahl in der internen Darstellung (entspricht sechs Byte).

Das Token »03« ist in unserem Beispiel bereits identifiziert. Es beginnt jetzt der Test, in welcher Reihenfolge die Operationen ablaufen müssen. Der Interpreter hat dabei fast die gleichen Tests durchzuführen, und zwar jedesmal, wenn er diese Zeile erreicht. Der Compiler übersetzt jede Zeile nur einmal bei der Compilierung. Dadurch laufen die Programme schneller ab.

Bei der Übersetzung in P-Code bedient sich der Compiler einer Tabelle, in der die Prioritäten aller Operatoren vermerkt sind. Übrigens handelt es sich hierbei fast um die gleiche Tabelle, die auch der Interpreter benutzt. Daraus ergibt sich dann folgender P-Code. In Bild 4 sind die einzelnen Werte lediglich in Kurzform dargestellt. Der Compiler betrachtet sie natürlich als Zahlen.

Jetzt könnte theoretisch schon die Routine aufgerufen werden, die den Maschinencode erzeugt. Allerdings würden dann im compilierten Programm die gleichen Berechnungen ablaufen, die der Interpreter normalerweise auch durchführt. Schließlich steht jetzt viel Zeit zur Optimierung des P-Codes zur Verfügung, da die Programmzeilen nur ein einziges Mal in Maschinencode umgesetzt werden.

Hier nochmals der P-Code:

```
Zahl 41 02 00 00 00 00
fp->intpeek int->fp push
Zahl 40 01 00 00 00 00
movepull add fp->int
```

Im ersten Schritt faßt der Optimierer die Zahl »41 02 00 00 00 00 fp->int« zusammen, indem er den Wert in die Integer-Darstellung umwandelt. Daraus entsteht:

```
Izahl 200 peek int->fp push
Zahl 40 01 00 00 00 00
movepull add fp->int
```

Anmerkung: Es gibt zwei Fp-Accus, $D4 bis $D9 und $E0 bis $E5. Integerzahlen stehen dabei im Prozessorregister A und Y. Dabei enthält Y das höherwertige und A das niederwertige Byte.

Der Optimierer faßt jetzt »Izahl 200 peek« zu »Ipeek 200« zusammen. Das erste Beispiel würde »LDA #<200:LDY #>200:JSR PEEK« als Code erzeugen und das zweite Beispiel »LDA 200:LDY #0«. Jetzt haben wir also:

```
Ipeek 200 int->fp push
Zahl 40 0100 00 00 00
movepull add fp->int
```

Jetzt wird »push Zahl ... movepull add« zu »add# ...« zusammengefaßt. Daraus ergibt sich:

```
1) COLOR 1                    ;so sieht's aus
2) 03 0E 40 01 00 00 00 00    ;so speichert's der Interpreter
3) A9 01 85 C8                ;das soll daraus werden (LDA #1:STA $C8)
                              ;$C8=200 ist die Speicherzelle, in der sowohl
                              ;Interpreter als auch Compiler die Zeichenfarbe
                              ;für Plot ablegen
```

Bild 2. So wird der Befehl »COLOR« compiliert

```
03 46 3A 0E 41 02 00 00 00 00 2C 35 40 01 00 00 00 00 16
```

Bild 3. So sieht der Basic-Befehl »COLOR PEEK(200)+1« in Token umgesetzt aus

```
Ipeek 200 int->fp add
# 40 01 00 00 00 00 fp->int
```

Als nächstes bemerkt der Optimierer die Folge »int->fp Add# ... fp->int« und versucht diese Befehlsfolge zu vereinfachen. So spart man sich später eine Umwandlung. Da die Zahl kleiner als 256 ist, gelingt dies auch.

```
Ipeek 200 Iadd# 1
```

Jetzt hat der Optimierer seine Arbeit geleistet. Anschließend wird richtiger Maschinencode erzeugt.

Aus »Ipeek 200« entsteht »LDA 200:LDY #0«,
»Iadd# 1« wird zu »LDX #1:JSR IADD« (dieser Unterprogrammaufruf ist nötig, da sonst auch ein Overflow stattfinden könnte; Integerwert größer als 65535!)

Das war auch schon alles. An den Befehl COLOR wird noch »STA $C8« angehängt. Insgesamt ergibt sich also:

```
LDA 200:LDY # 0:LDX # 1:JSR
IADD:STA $C8
```

Ohne Optimierung würde wesentlich mehr Programmcode erzeugt werden, der noch dazu viel langsamer ist:

```
LDA #< Z200:LDY #>Z200:JSR
LDOYA:JSR FPI?:JSR PEEK:JSR
IFP:JSR PUSH
LDA # <Z1:LDY # > Z1:JSR
MOVEPULL:JSR ADD:JSR FPI?:STA $C8
```

Außerdem noch im Konstantenbereich:

```
Z200 .BYTE $41,$02,$00,$00,$00,
$00
Z1   .BYTE $40,$01,$00,$00,$00,
$00
```

Durch Optimierung läßt sich also viel Programmcode sparen. Gegenüber einem echten Maschinencode sieht das Ergebnis allerdings immer noch recht umfangreich aus. In Maschinensprache würde nämlich »INC $C8«, was nur 2 Byte benötigt, das gleiche bewirken. Hexadezimal $C8 entspricht dabei dem Dezimalwert 200. In unserem Beispiel codiert der Compiler den Basicbefehl COLOR 1 jedoch optimal, nämlich »LDA #1:STA $C8«.

Optimal compiliert

Allerdings kann der Compiler nur in den seltensten Fällen optimalen Code erzeugen. Schließlich erübrigt sich nur die Umwandlung von FP nach INT und umgekehrt und dies auch nur bei Verwendung von Konstanten. Bei Variablen ist diese Art der Optimierung nicht zulässig. In den meisten Fällen reicht schon die Einsparung einiger Bytes, weil zum Beispiel das Laden einer Variablen und die Umwandlung in einen statt zwei Integerwerte durch einen einzigen Unterprogrammaufruf erledigt wird.

Auch wird nicht alles, was möglich wäre, optimiert. Zum Beispiel berechnet sich »A=10/3« nicht im voraus, obwohl es durchaus sinnvoll wäre. Aber solche Optimierungen kann der Programmierer gegebenenfalls selbst vorwegnehmen. Allerdings sind solche Fälle viel zu selten, als daß sich dies hier lohnen würde. Je länger nämlich der Compiler, desto kürzer die entsprechenden Basic-Programme. Es gibt jedoch einige Berechnungen, deren Optimierung sich besonders lohnt: »^2« erhält eine eigene, schnelle Routine. Im Compiler benötigt »A=B^2« die gleiche Zeit wie »A=B*B«. Es existiert ebenfalls eine schnelle Exponential-Routine, die bereits im Mathematik-Teil der Runtime-Bibliothek vorhanden ist. So oder ähnlich werden alle Befehle mit Parametern behandelt. Sind mehrere Parameter vorhanden, werden diese natürlich zwischengespeichert.

Ein weiterer lohnenswerter Befehl ist

```
Zahl 41 02 00 00 00 00      ;die 200
fp->int peek int->fp        ;der PEEK-Befehl benötigt einen Integer als Adresse,
                            ;deshalb muß die 200 zuerst umgewandelt werden.
                            ;Dann folgt der eigentliche PEEK-Befehl und dann
                            ;die Umwandlung des Ergebnisses in eine
                            ;Floating-Point-Zahl (fp). Diese
                            ;Umwandlungen muß der Interpreter auch immer
                            ;durchführen.
push                        ;retten des Ergebnisses auf den Rechenstapel
Zahl 40 01 00 00 00 00      ;Zahl 1
Movepull                    ;Schiebt die Zahl 1 aus dem FP-Accu 1 in den
                            ;FP-Accu 2 und holt das Ergebnis des PEEKs
                            ;in den FP-Accu 1
Add                         ;die Addition
fp->int                     ;dies wird von der Compilationsroutine für Color
                            ;angehängt, es wird ja ein Integer benötigt
```

Bild 4. Der P-Code, der sich aus »COLOR PEEK(200)+1« ergibt

der POKE-Befehl. Beispielsweise wird »POKE 16,64« durch »LDA #64:STA 16« übersetzt.

Beim PRINT-Befehl sind noch die Kommata zu beachten. Auch INPUT, READ, GET und LOCATE beinhalten noch einige Probleme. Eine ausführliche Erläuterung würde den Rahmen dieses Artikels jedoch sprengen.

Gesondert zu behandeln sind alle Befehle, die den linearen Programmablauf beeinflussen. Der GOTO-Befehl läßt sich vergleichsweise einfach übersetzen. Zuerst wird die Routine aufgerufen, die einen Integer-Ausdruck übersetzt und optimiert (wie oben); allerdings generiert sie ihn noch nicht in den endgültigen Maschinencode. Zunächst wird nur der P-Code erzeugt. Dann wird anhand des P-Codes überprüft, ob eine Zahl folgt. Wenn ja, wird ein JMP-Befehl compiliert. Zunächst nur »07 Zeile«, woraus sich nach Paß 2 »4C Adresse« ergibt. Wenn nein, wird die Routine für die Erzeugung von Maschinencode aufgerufen und »JSR GOTOX« angehängt. Bei einem berechneten GOSUB wird beispielsweise entweder »JSR GOSUBX« oder vor dem »JMP-($07)«-Befehl der GOTO-Anweisung ein Unterprogrammaufruf compiliert. Dieser enthält dann, auf dem Basic-Stack, die um drei Werte erhöhte Rücksprungadresse des Befehls. Es wird aber nicht einfach ein JSR compiliert, da sonst die Anzahl der Unterprogrammebenen durch den zu kleinen 6502-Stack begrenzt wäre. Außerdem müssen die Parameter für die FOR-NEXT-Schleifen ebenfalls auf diesem Stapel abgelegt werden. Sie belegen jeweils 16 Byte, so daß sehr schnell ein Overflow des 6502-Stack drohen würde. Dies könnte katastrophale Folgen haben und die Kompatibilität zum Interpreter-Basic stark einschränken. Zumindest ist die untere Hälfte der Seite 1 frei, und es gibt Programme, die mehr als 10 geschachtelte Schleifen verwenden. Der Zeitverlust durch diesen simulierten Stack hält sich jedoch in Grenzen.

Einen ähnlichen Stack verwendet der Compiler auch zur Übersetzung von DO-LOOP, REPEAT-UNTIL, WHILE-WEND und FOR-NEXT-Schleifen sowie des EXIT-Befehls.

Bei DO wird die aktuelle Adresse auf diesem Stapel abgelegt. Sie erhält außerdem noch die DO-Befehlskennung. Bei Übersetzung des zugehörigen LOOP wird die Kennung daraufhin überprüft, ob ein DO fehlt und dann ein JMP-Befehl auf die zuvor festgestellte Adresse compiliert. Sollte zwischen DO und LOOP ein EXIT gestanden haben, so hat dieser Befehl seine eigene Kennung auf dem Stack hinterlassen und einen JMP-Befehl mit »Dummy«-Adresse erzeugt. LOOP ersetzt diese Adresse durch die auf den eigenen JMP-Befehl folgende.

Geschwindigkeit ist Trumpf

Bei REPEAT ist der Vorgang ähnlich wie bei DO. Es unterscheidet sich lediglich die Kennung. Wenn UNTIL auf eine REPEAT-Anweisung folgt, wird zuerst der folgende Ausdruck übersetzt. Erst dann wird ein bedingter Sprung angehängt. Da der 6502 nur relativ kurze, bedingte Sprünge kennt, wird mit »BNE *+5:JMP ad« gearbeitet. Natürlich muß auch UNTIL eventuelle EXIT-Befehle berücksichtigen.

WHILE-WEND entspricht dem eben Besprochenen, nur steht hier die Bedingung am Schleifenanfang.

Bei FOR-NEXT-Schleifen ist noch zu beachten, daß zusätzlich der Basic-Stack benutzt wird. Deshalb wird automatisch ein POP-Befehl in den EXIT-Befehl eingeschlossen.

Damit ist die Wirkungsweise des Compilers erklärt. Vielleicht ist Ihnen aber noch nicht ganz klar, wie der Interpreter und der Compiler die Reihenfolge der Berechnungen feststellen. Der Interpreter führt die Berechnungen aus, der Compiler erzeugt einen P-Code. Trotzdem funktionieren beide nach dem gleichen Prinzip. Zur Zwischenspeicherung der Operatoren dient jeweils ein Stack (Stapel). Dazu ein einfaches Beispiel:

```
8*Sin(A)+16
```

Der Compiler liest die Ziffer 8. Als Zahl wird sie sofort compiliert. (Der Interpreter würde sie auf dem Zahlenstapel ablegen.) Dann wird das Token für Multiplikation (*) mit der Priorität 5 gelesen. Da der Stapel für Operatoren noch leer ist (Priorität 0), wird das Token dort abgelegt. Als nächstes folgt SIN mit der höchstmöglichen Priorität (9). Weil diese größer ist als die auf dem Stapel (5), wird auch SIN auf den Stapel gelegt. Dann folgt die geöffnete Klammer. Als Spezialfunktion ruft die Klammer dieselbe Routine noch einmal, allerdings rekursiv auf. Es wird also der Wert in Klammern entsprechend übersetzt oder berechnet. Dann kommt die Variable A an die Reihe. Variablen werden, genauso wie Zahlen, sofort compiliert. Dann schließt als Endekennzeichen die Klammer ab. Weiter geht es mit der ursprünglichen Übersetzung. Es folgt nun das Token für Addition (+), dessen Priorität (4) kleiner als die auf dem Stapel ist. Deshalb wird jetzt der SIN-Befehl compiliert. Anschließend wird noch die vorliegende Priorität mit der des »*«-Tokens (5) verglichen. Da sie wieder kleiner ist, wird jetzt die Multiplikation compiliert.

Der leere Stack hat die Priorität 0, also wird das Token für Addition auf dem Stack abgelegt. Dann folgt die Zahl 16 und schließlich das Endekennzeichen mit der Priorität 0. Der Stapel ist anschließend leer, die Addition wird compiliert, und der Vorgang ist beendet. Der entstandene P-Code lautet »Zahl 8 Var A Sin Mult Zahl 16 Add«.

So einfach wie hier beschrieben, ist das Compilieren von Basic-Programmen jedoch nicht. Es sollte an dieser Stelle vielmehr gezeigt werden, wie der Turbo-Compiler prinzipiell funktioniert. Sollten alle Vorgänge beim Compilieren beschrieben werden, würde dies ein Buch füllen. Noch ein Tip: Das Standard-Atari-Basic ist wegen eines Betriebssystem-Fehlers nicht in der Lage, »A=NOT NOT A« zu berechnen. Dies hängt mit der Stapelverarbeitung zusammen. Probieren Sie diese Befehlsfolge doch einfach einmal auf Ihrem Atari 800XL aus. Aufgrund der Syntaxkontrolle ist der Befehl nicht zulässig. Die alten Atari-Computer reagierten auf »A=NOT NOT A« mit einem Systemabsturz. Geben Sie die Programmzeile doch einfach einmal unter Turbo-Basic-XL ein. Sie werden sehen, daß auch solche komplexen, logischen Verknüpfungen ordnungsgemäß funktionieren. (Frank Ostrowski/wb)

Bücher zum ATARI 65XE/800XL/130XE

L. M. Schreiber
Das Atari-Programmierhandbuch
März 1985, 390 Seiten

BASIC lernen, seinen Atari – einschließlich Maschinensprache – bis ins Detail kennenlernen und am Schluß eine Menge Programme haben, mit denen man vor seinen Freunden so richtig angeben kann (wessen Computer kann schon eine dreistimmige Fuge von J. S. Bach spielen?): Wenn Sie das wollen, dann ist dieses Buch für Sie gerade richtig, auch oder gerade wenn Sie Anfänger sind. Nach vollendeter Lektüre wird man das Buch aber nicht zur Seite legen: Es enthält eine solche Fülle an Informationen, daß es auch als Nachschlagewerk bestens geeignet ist.
• BASIC- und Atari-Einführung für den Anfänger; Nachschlagewerk für den Fortgeschrittenen.
Best.-Nr. MT 753
ISBN 3-89090-062-3
DM 52,–/sFr. 47,80/öS 405,60





L. Poole/M. McNiff/S. Cook
Mein Atari-Computer
1983, 500 Seiten

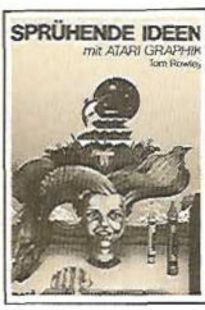
Ein Handbuch, das für jeden Atari-Besitzer wertvolle Informationen enthält. Ob Sie einen Atari 400, 800, 600 XL, 800 XL oder 1450 XLD besitzen, das Handbuch gibt Ihnen Auskunft zu allen Fragen der Bedienung und BASIC-Programmierung, zum Umgang mit Kassetten, Drucker und Diskettenstation, zum Erzeugen von Bild und Ton. Es ist reich bebildert und enthält eine Vielzahl der für den ernsthaft Interessierten so wichtigen Tabellen. Besonders im Anhang findet sich so mancher Leckerbissen für den echten Freak, z.B. eine Erklärung der wichtigsten Speicheradressen für PEEKs und POKEs.
• Ein Handbuch zur Lösung aller Atari-Probleme.
Best.-Nr. PW 554
ISBN 3-921803-18-7
DM 59,–/sFr. 54,30/öS 460,20

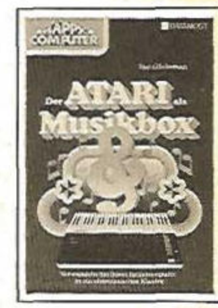












H. L. Schneider/R. Bichler
Das Atari-Buch
Band 1: Grundlegende Programmiermöglichkeiten
Oktober 1984, 158 Seiten

Auch wer BASIC kann und sein Atari-Handbuch durchgearbeitet hat, wird ohne zusätzliche Anregungen seinen Atari-Computer nicht völlig ausschöpfen können.
Best.-Nr. MT 703
ISBN 3-89090-039-9
DM 32,–/sFr. 29,50/öS 249,60

Band 2: Spezielle Programmiermöglichkeiten und Maschinenprogramme
Oktober 1984, 197 Seiten

Das ideale Buch für die Atari-Programmierung in Maschinensprache.
Best.-Nr. MT 704
ISBN 3-89090-072-0
DM 32,–/sFr. 29,50/öS 249,60

E. H. Carlson
Lerne BASIC auf dem Atari
November 1984, 320 Seiten

Zu einem Computer für junge Leute gehört auch ein Lehr- und Lernbuch für junge Leute. Da tut es nicht irgendein BASIC-Buch: der Aufbau, die verwendeten Beispiele, das Lerntempo und nicht zuletzt der Ton müssen stimmen. Nichts schreckt mehr ab als schauderhafter Oberlehrer-Schulbuchstil! So drehen sich auch die Beispiele vorzugsweise um Spiele (Action-, Brett- und Wortlisten), denn auch an ihnen läßt sich Programmieren erlernen. Die Lektionen sind nicht zu lang, mit Grafiken aufgelockert und mit Übungsbeispielen versehen (selbstverständlich inklusive Lösungen).
Best.-Nr. MT 692
ISBN 3-89090-007-0
DM 38,–/sFr. 35,–/öS 296,40

H. Kohl/T. Kahn/
L. Lindsay/P. Cleland
Spiel und Spaß mit dem Atari
1984, 338 Seiten

Den Umgang mit dem Computer lernt man am besten durch Ausprobieren; deshalb sollte ein Einführungsbuch möglichst viele Beispiele enthalten. In diesem Buch finden Sie daher über 100 Programmbausteine, die zusammen mit 20 fertigen und vollständigen Programmen ein solides Fundament für den Umgang mit BASIC und Ihrem Computer legen. So können Sie besser die interessanten Ton- und Grafikmöglichkeiten des Atari nutzen.
• Ein Lehr- und Arbeitsbuch für Anfänger, die BASIC und ihren Atari kennenlernen wollen.
Best.-Nr. MT 672
ISBN 3-89090-002-X
DM 42,–/sFr. 38,60/öS 327,60

A. Z. Lamothe jr.
Ausgesuchte Atari-Programme mit Listings
Oktober 1984, 171 Seiten

Es gibt drei Möglichkeiten, mit einem Computer zu arbeiten: man kann fertige Programme kaufen, man lernt selber programmieren oder tippt Programme ab und macht sich so mit der Funktion seines Computers vertraut.
Dieses Buch ist für diejenigen gedacht, die es mit der dritten Methode versuchen wollen. Durch das Abschreiben der Programme lernen Sie leichter den Umgang mit Ihrem Computer und werden an die Grundbegriffe des Programmierens herangeführt.
Alle Programme laufen mit der Atari-BASIC-Version, die mit fast allen Atari-Computern ausgeliefert wird.
Best.-Nr. MT 759
ISBN 3-89090-070-4
DM 32,–/sFr. 29,50/öS 249,60

T. Rowley
Sprühende Ideen mit Atari-Grafik
Januar 1985, 224 Seiten

Ein eigenes Buch nur für Grafik? Bei den umwerfenden Grafik-Möglichkeiten des Atari ist es das schon wert. Schließlich wollen Sie ja nicht nur in die Grundlagen der Bilderzeugung eingeführt werden: Es soll schon auch bunt, bewegt und dreidimensional zugehen. Wie man Zeichensätze verändert, erfahren Sie selbstverständlich auch. Ein bißchen BASIC sollten Sie aber schon können, um möglichst viel von diesem Buch zu haben.
• Grafikprogrammierung für BASIC-Erfahrene auf dem Atari 400 oder 800.
Best.-Nr. PW 716
ISBN 3-921803-39-X
DM 49,–/sFr. 45,10/öS 382,20

H. Glicksman
Der Atari als Musikbox
November 1984, 194 Seiten

Die Soundmöglichkeiten des Atari sind schon beinahe legendär; sie nutzen Ihnen jedoch wenig, wenn Sie über die Prinzipien der Klangerzeugung und Musikprogrammierung nicht Bescheid wissen. Deshalb bringt dieses Buch eine Fülle einfacher und lehrreicher Programmbeispiele, die auch dem in BASIC nicht so Versierten das Verständnis der wichtigsten Techniken erlauben. Und so können Sie zuletzt nicht nur beliebige Klänge erzeugen oder den Computer in einen Synthesizer verwandeln, Sie besitzen auch eine eigene »Diskothek« von beliebten und bekannten Melodien, die Ihr Atari auf Knopfdruck vierstimmig spielen kann.
Best.-Nr. MT 797
ISBN 3-89090-075-5
DM 29,80/sFr. 27,50/öS 232,40

Markt & Technik-Fachbücher erhalten Sie bei Ihrem Buchhändler

Bestellkarten bitte an Ihren Buchhändler oder an einen unserer Depot-Händler. Adressenverzeichnis am Ende des Heftes.

Bestellungen im Ausland bitte an untenstehende Adressen.
Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, ☎ 042/41 56 56
Österreich: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, ☎ 0222/67 75 26



Hans-Pinsel-Straße 2, 8013 Haar bei München



Der Unterschied liegt im Detail

Läuft Ihr Programm nicht unter Turbo-Basic XL? Einige Änderungen lassen auch Ihr normales Atari-Basic-Programm zum Renner werden.

Turbo-Basic XL ist wesentlich schneller als das eingebaute Atari-Basic (siehe hierzu auch den Geschwindigkeitsvergleich zwischen den derzeit erhältlichen Basic-Versionen für den Atari 800XL und 130XE in diesem Sonderheft). Die hohe Geschwindigkeit ist in den meisten Anwendungsfällen sicher erwünscht. Allerdings ergeben sich daraus auch manchmal Probleme. Oft sind nämlich in Atari-Basic geschriebene Spielprogramme unter Turbo-Basic XL kaum mehr spielbar, da die enorme Geschwindigkeit dem Spieler nicht mehr genügend Reaktionszeit erlaubt. Abhilfe: Verzögerungen in Form von Warteschleifen einbauen oder den speziellen PAUSE-Befehl verwenden. Oder bauen Sie doch einfach zusätzliche Geräusch-Effekte ein.

In Atari-Basic wird eine kurze Verzögerung auch mit »Q=1^1« realisiert. In Turbo-Basic XL wird diese Berechnung aber wesentlich schneller als in Atari-Basic ausgeführt. Potenzierung mit ganzzahligem Exponenten (<100) wird durch wiederholte Multiplikation berechnet, und nicht, wie in Atari-Basic, nach der Formel »a^b=exp(b*log(a))«. Am einfachsten ist es, »Q=1^1« durch »PAUSE 9« zu ersetzen. Dies nimmt etwa die gleiche Zeit in Anspruch und spart sogar etwas Speicherplatz.

Verzögerungsschleifen mit FOR.. NEXT.. sind in Atari-Basic sehr langsam – besonders wenn sie sich am Programmende befinden. Am besten ersetzen Sie solche Schleifen auch durch entsprechende PAUSE-Befehle.

Verbotene Befehle

Turbo-Basic XL beinhaltet wesentlich mehr Befehle als Atari-Basic. Deshalb können Probleme mit Variablennamen auftreten. Namen wie ERR, HEX$ oder DEL(ta) sind nämlich in Atari-Basic erlaubt, in Turbo-Basic XL jedoch nicht (es handelt sich hierbei um reservierte Worte). Beim Laden und Starten eines Atari-Basic-Programms macht sich dieser Unterschied noch nicht bemerkbar. Erst wenn Sie versuchen, ein Programm mit diesen Variablennamen zu editieren, ergeben sich Fehlermeldungen oder falsch übersetzte Programmzeilen. Bei »LET ERR=100« wird die Variable ERR benutzt, bei »A=ERR« jedoch die Turbo-Basic XL-Funktion »ERR«. Dann bleibt als einzige Alternative, die Variablennamen manuell zu ändern. Dabei hilft übrigens der DUMP-Befehl, mit dem man sich die im Programm verwendeten Variablen auflisten lassen kann.

Nach dem Laden des betreffenden Programms mit LOAD geben Sie einfach DUMP ein, und alle Variablen werden auf dem Bildschirm ausgegeben. Falls Sie einen Drucker besitzen, erfolgt die Ausgabe nach »DUMP "P:"« auf dem Drucker. Schreiben Sie sich dann alle Variablennamen auf, die geändert werden müssen. Dabei bleibt es Ihnen nicht erspart, jede einzelne Programmzeile zu kontrollieren und gegebenenfalls zu ändern. Speichern Sie das Programm anschließend mit »LIST "D:xxx.xxx"« auf Diskette, geben Sie NEW und dann »ENTER "D:xxx.xxx"« ein. Diese Befehlsfolge bewirkt, daß die Variablennamentabelle neu geschrieben wird. Nicht benutzte Variablen belegen dann auch keinen Speicherplatz mehr. Mit DUMP können Sie nochmals kontrollieren, ob Sie irgendwelche Variablen übersehen haben. Wenn Sie alle Korrekturen vorgenommen haben, sichern Sie das Programm mit SAVE auf Diskette und führen Sie einen Probelauf durch.

Turbo-Basic XL liegt nicht im ROM vor, sondern wird von Diskette geladen. Da es noch wesentlich umfangreicher als Atari-Basic ist (etwa 18 KByte statt 8 KByte), wurde ein Großteil des Interpreters unter den OS-ROMs versteckt (12 KByte). Die restlichen 6 KByte befinden sich an der Speicheruntergrenze direkt über dem DOS.

Ursprünglich wurde Turbo-Basic XL auf einem Atari 400 mit 48 KByte RAM geschrieben. Bei dieser Version befand sich der gesamte Interpreter über dem DOS. Allerdings blieben so nur etwa 20 KByte für Basic-Programme übrig. Der 800XL besitzt dagegen insgesamt 64 KByte RAM, die Turbo-Basic XL auch voll nutzt. Von den zusätzlichen 14 KByte werden aber noch 2 KByte von Kopien der beiden ROM-Zeichengeneratoren belegt. Eine Maßnahme, die Bildstörungen bei Textdarstellung – sobald das ROM abgeschaltet wird – verhindert. Turbo-Basic XL ist nun aber viel zu umfangreich, um in die verbleibenden 12 KByte hineinzupassen. Deshalb ließ es sich nicht umgehen, auch einige Programmteile in Bereichen abzulegen, die normalerweise für Basic-Programme reserviert sind. Auf diese Art und Weise lassen sich Unterprogramme der ROMs benutzen. Weiterhin erlaubt dies noch die Durchführung der Interrupts.

Der im Basic-Bereich untergebrachte Teil von Turbo-Basic XL belegt die Adressen von $2080 bis $3629 (hexadezimal). Es handelt sich also um den Bereich direkt oberhalb von DOS 2.5 bei fünf installierten Diskettenlaufwerken (D1: bis D4: und der RAM-Disk D8: auf dem 130 XE) und 7 Filepuffern. Ein POKE in diesen Bereich dürfte meist zum Absturz des Computers führen, oder – was weitaus schlimmer wäre – zu einem Fehlverhalten des Interpreters. Denn daraus können sich sogar falsche Rechenergebnisse ableiten.

Während der Ausführung der Befehle POKE, DPOKE, MOVE und -MOVE wird jeweils der ROM-Bereich eingeschaltet. So können diese Befehle im »ROM«-Teil von Turbo-Basic XL keinen Schaden anrichten. Außerdem ist der ROM-Bereich noch bei den Befehlen PEEK, DPEEK, allen I/O-Befehlen wie PRINT, INPUT, BGET, BPUT etc. aktiviert, jedoch nicht bei PAINT, sowie der Ausführung einiger Arithmetik-Routinen (es werden einige Werte aus den Floating-Point-Routinen sowie die STR$- und VAL-Routinen verwendet).

Vorsicht, Interrupt!

Tritt ein Interrupt auf, während die ROMs eingeschaltet sind, wird er wie gewohnt ausgeführt. Wenn die ROMs allerdings ausgeschaltet sind, verzweigt der Interrupt zuerst in den unteren RAM-Teil des Interpreters. Dort werden zuerst die ROMs eingeschaltet. Nachdem ein Interrupt für die ROM-Routinen simuliert wird, werden die ROMs wieder stillgelegt. Aus diesem Grund dauern die Interrupt-Routinen einige Mikrosekunden länger. Dies gilt allerdings nicht für den Display-List-Interrupt (DLI), da dieser parallel zum Bildaufbau ablaufen muß. Der Interrupt wird hier schon vom Interpreter erkannt, so daß er sofort ausgeführt werden kann. Er muß also auch bei abgeschalteten ROMs ablaufen.

Der Speicherplatz für Basic-Programme, Daten und den Stack reicht

unter Atari-Basic bis $A000 (40960), unter Turbo-Basic XL aber bis $C000 (49152). Das Basic-ROM wird hierbei nämlich nicht benutzt und kann deshalb immer abgeschaltet bleiben. Reine Basic-Programme, also ohne USR-Aufrufe und POKEs, sind von dieser Verschiebung nicht betroffen. Kritisch sind POKEs und USR-Aufrufe in Adressen im Bereich zwischen 30000 und 40960. Wenn Sie also ein Programm verwenden, das einen Befehl wie beispielsweise »POKE 39994,4« enthält, dürfte das Programm unter Turbo-Basic XL nur mit einigen Anpassungen laufen. Dieser POKE-Befehl verändert übrigens die Display-List. Hierbei handelt es sich um die Maschinensprache des Videoprozessors, der Antic genannt wird.

Da es auch noch den 600 XL mit 16 KByte und den Atari 400 und 800 mit 16, 32 und 48 KByte RAM gibt, existieren auch Programme, die sich an verschiedene Speichermengen anpassen. Dazu benutzt das OS (Operating System) des Atari verschiedene Pointer (Zeiger), auf die auch Basic zurückgreift. Der wichtigste davon ist RAMTOP, also die Speicherzelle 106 ($6A). Sie enthält das MSB (Most Significant Byte) der ersten nicht mehr benutzbaren Speicherzellen für Programme. Das heißt, nach »PEEK(106)*256« erhält man einen Wert, der um eins höher ist als die höchste RAM-Adresse. In Atari-Basic ergibt »PEEK(106)« also 160 ($A0), in Turbo-Basic XL entsprechend 192 ($C0). Durch »POKE 106,PEEK (106)-16« lassen sich so 4096 Bytes (16*256) vor dem Zugriff durch das OS und Basic schützen.

Anschließend ist noch ein GRAPHICS-Befehl nötig, um die anderen Pointer anzupassen und um den Bildschirmspeicher zu verschieben. Durch »POKE 106,160:GRAPHICS 0« am Anfang eines Programms wird also die Atari-Basic-Speicheranordnung simuliert. Dann bleiben allerdings wieder 8 KByte weniger Speicherplatz für das Basic-Programm übrig. Dafür beginnt aber der Bildschirmspeicher schon ab Adresse 40000 statt 48192. Für kurze Programme reicht diese Minimalanpassung vollkommen aus. Sollte ein Programm aber länger sein, wie zum Beispiel »Magic-Painter« (Listing des Monats, Happy-Computer, Ausgabe 3/85), gibt es zwei Möglichkeiten:

Adresse	Atari-Basic
FFFF	OS-ROM
D800	I/O-Chips
D000	OS-ROM
C000	Basic-ROM
A000	Grafik-RAM
	Basic-Stack
	Arrays
	Programm
2080	DOS 2.5
0700	OS-Variablen
0000	

Adresse	Turbo-Basic XL
FFFF	OS-ROM / Interpreter
D800	I/O-Chips
D000	OS-ROM / Interpreter
C000	Grafik-RAM
	Basic-Stack
	Arrays
	Programm
3629	Interpreter
2080	DOS 2.5
0700	OS-Variablen
0000	

Die Speicheraufteilung von Turbo-Basic XL im Vergleich zum normalen Atari-Basic.

1) Alle POKE- und USR-Adressen um 8192 erhöhen. Dies klappt aber nur bei relokatiblen, das heißt, bei verschiebbaren Daten und Maschinenprogrammen. Leider sind diese recht selten; und wenn sie autreten, dann hat der Programmierer das Maschinenprogramm meist in einem String untergebracht (das Kennzeichen dafür ist stets: »USR(ADR(X$),...))«. Dieser Aufruf bereitet aber keine Schwierigkeiten. Sie können natürlich die Maschinenprogramme relozieren, also an die andere Adreßlage anpassen. Dies erfordert aber Assemblerkenntnisse und viel Zeit- und Arbeitsaufwand. Einfacher ist es, wenn der Quelltext des Programms zur Verfügung steht.

Verbotene Adressen

Auf Probleme stößt man auch bei den Daten für Player/Missile-Grafiken oder bei veränderten Zeichensätzen. Hier sind die POKE-Befehle, die deren Adressen dem Operating System oder der Hardware mitteilen, anzupassen. Dabei handelt es sich meist um POKEs auf die Adressen 756, 54279, 561 und 89 (und 513 etc.). Der gePOKEte Wert ist dann jeweils um 32 zu erhöhen. Manchmal werden diese Adressen auch innerhalb von Maschinenprogrammen angesprochen.

Sie sehen also, daß diese Anpassung ohne genaue Kenntnisse des Programms und der Hardware fast unmöglich ist.

2) Das Programm um 8 KByte kürzen. Dies ist bei großzügig (gut) kommentierten Programmen manchmal allein durch Streichen der REM-Zeilen zu erreichen. Oft hilft auch das Auslagern von DATAs in ein Vorprogramm, das in einer READ-Schleife Daten einPOKEt, und dann das gekürzte Hauptprogramm mit »Run "D:xxx.xxx"« startet. Schneller ist aber ein entsprechender BLOAD- oder BGET-Befehl, wenn die Daten im entsprechenden Format auf Diskette vorliegen. Dies spart übrigens auch wertvollen Platz auf der Diskette. Weiterhin läßt sich auch durch geschickten Einsatz der erweiterten Turbo-Basic XL-Befehle viel Speicherplatz sparen.

Durch einen kombinierten Einsatz dieser Mittel wurde auch das Programm »Magic-Painter« angepaßt. Besonders hilfreich waren dabei die Befehle BLOAD, CIRCLE und TEXT. Noch ein Tip: Wäre »Magic-Painter« in Turbo-Basic XL geschrieben worden, könnte es ohne weiteres in der höchstauflösenden Farbgrafikstufe 15+16 arbeiten, statt in Grafikstufe 7+16. Schließlich stehen unter Turbo-Basic XL 8 KByte RAM mehr zur Verfügung.

(Frank Ostrowski/wb)

Auf die Taste, fertig, los!

Wie schnell Turbo-Basic-XL und der Turbo-Basic-Compiler wirklich sind, soll ein Vergleich mit anderen Programmiersprachen und Compilern zeigen.

Um Aussagen über die Geschwindigkeit eines Computers oder einer Programmiersprache machen zu können, bedient man sich üblicherweise sogenannter Benchmark-Programme. Hierbei handelt es sich um kurze Algorithmen, die möglichst ohne Änderung auf allen zu testenden Versionen laufen sollten. In diesem Fall wurden alle Benchmarks in Standard-Atari-Basic geschrieben und, soweit dies durchführbar war, direkt in die anderen Sprachen übernommen.

Neun unterschiedliche Benchmarks werden erweisen, wie schnell Turbo-Basic XL und der Turbo-Compiler im Vergleich mit anderen Programmiersprachen für den Atari sind. Zur Zeitmessung wurden die Speicherstellen 19 und 20 herangezogen. Die Speicherstelle 20 wird durch einen Interruptimpuls 50mal pro Sekunde inkrementiert.

Ist der Wert 256 erreicht, erhöht sich der Wert in Speicherstelle 19 um 1 und Adresse 20 wird auf 0 zurückgesetzt. Liest man diesen Zwei-Byte-Wert und teilt ihn durch den Faktor 50, so erhält man eine Zeitangabe in Sekunden. Um die Unterschiede der Rechenzeiten deutlicher zu machen, wird jeder Benchmark 1000mal durchlaufen.

Als erste Vergleichssprache diente natürlich das normale Atari-Basic. Wie man der Tabelle entnehmen kann, ist

```
10 DIM M(5),A$(1000)                <CC>
20 ? "ATARI▲BENCHMARKS"              <CH>
99 REM *** Bench 1 ***               <PA>
100 POKE 20,0:POKE 19,0              <XQ>
110 FOR I=1 TO 1000                  <GV>
120 NEXT I                           <FV>
130 T1=PEEK(20)+PEEK(19)*256         <CE>
190 ? "T1▲=▲";T1/50                  <GZ>
199 REM *** Bench 2 ***              <OZ>
200 POKE 20,0:POKE 19,0              <XR>
210 K=0                              <FP>
220 K=K+1                            <RD>
230 IF K<1000 THEN 220               <QG>
240 T2=PEEK(20)+PEEK(19)*256         <CN>
290 ? "T2▲=▲";T2/50                  <HZ>
299 REM *** Bench 3 ***              <PT>
300 POKE 20,0:POKE 19,0              <XS>
310 K=0                              <FQ>
320 K=K+1                            <RE>
330 A=K/K*K+K-K                      <QH>
340 IF K<1000 THEN 320               <RD>
350 T3=PEEK(20)+PEEK(19)*256         <CW>
390 ? "T3▲=▲";T3/50                  <IZ>
399 REM *** Bench 4 ***              <QN>
400 POKE 20,0:POKE 19,0              <XT>
410 K=0                              <FR>
420 K=K+1                            <RF>
430 A=K/2*3+4-5                      <ZI>
440 IF K<1000 THEN 420               <RY>
450 T4=PEEK(20)+PEEK(19)*256         <DD>
490 ? "T4▲=▲";T4/50                  <JZ>
499 REM *** Bench 5 ***              <RH>
500 POKE 20,0:POKE 19,0              <XU>
510 K=0                              <FS>
520 K=K+1                            <RG>
530 A=K/2*3+4-5                      <ZJ>
540 GOSUB 2000                       <RF>
550 IF K<1000 THEN 520               <SV>
560 T5=PEEK(20)+PEEK(19)*256         <DM>
590 ? "T5▲=▲";T5/50                  <KZ>
599 REM *** Bench 6 ***              <SB>
600 POKE 20,0:POKE 19,0              <XV>
610 K=0                              <FT>
620 K=K+1                            <RH>
630 A=K/2*3+4-5                      <ZK>
640 GOSUB 2000                       <RG>
650 FOR L=1 TO 5                     <PD>
660 NEXT L                           <HM>
670 IF K<1000 THEN 620               <TU>
680 T6=PEEK(20)+PEEK(19)*256         <DX>
690 ? "T6▲=▲";T6/50                  <LZ>
699 REM *** Bench 7 ***              <SV>
700 POKE 20,0:POKE 19,0              <XW>
710 K=0                              <FU>
720 K=K+1                            <RI>
730 A=K/2*3+4-5                      <ZL>
740 GOSUB 2000                       <RH>
750 FOR L=1 TO 5                     <PE>
755 M(L)=A                           <QM>
760 NEXT L                           <HN>
770 IF K<1000 THEN 720               <UP>
780 T7=PEEK(20)+PEEK(19)*256         <EE>
790 ? "T7▲=▲";T7/50                  <MZ>
799 REM *** Bench 8 ***              <TP>
800 POKE 20,0:POKE 19,0              <XX>
810 K=0                              <FV>
820 K=K+1                            <RJ>
830 A=K^2                            <FO>
840 B=LOG(K)                         <WU>
850 C=SIN(K)                         <ZN>
860 IF K<1000 THEN 820               <VI>
870 T8=PEEK(20)+PEEK(19)*256         <EJ>
890 ? "T8▲=▲";T8/50                  <NZ>
899 REM ***BENCH 9 ***               <KG>
900 POKE 20,0:POKE 19,0              <XY>
910 FOR I=1 TO 1000                  <HD>
920 A$(I,I)="A"                      <WZ>
930 NEXT I                           <GF>
940 T9=PEEK(20)+PEEK(19)*256         <EK>
950 ? "T9▲=▲";T9/50                  <OR>
1000 TS=T1+T2+T3+T4+T5+T6+T7+T8+T9   <MU>
1010 ? :? "SUM=▲";TS/50              <HA>
1020 END                             <PV>
2000 RETURN                          <OU>
```

Listing zu »Benchmarks«

Benchmark Nummer	Atari-Basic	Turbo-Basic	Basic XE normal	Basic XE fast	Turbo-Compiler	Action	MMG-Compiler	ABC-Compiler
T1	2.22	0.76	1.50	1.42	0.50	0.02	0.92	0
T2	7.64	3.06	4.56	3.04	0.54	0.02	1.14	1
T3	20.24	7.58	16.08	8.82	2.82	0.82	8.38	5
T4	24.12	8.62	16.66	9.72	3.78	16.14	11.86	5
T5	33.74	9.08	25.92	17.62	3.79	16.14	11.86	6
T6	58.30	14.54	45.40	37.12	7.00	16.26	16.44	10
T7	83.58	24.86	62.28	49.98	11.78	17.86	22.73	13
T8	425.34	56.02	419.80	59.28	50.30	467.44	417.22	—
T9	8.14	3.24	6.88	4.20	1.24	0.02	2.68	1
T1–T7	229.84	68.50	172.40	127.72	30.38	67.26	73.32	40
Summe	663.32	127.76	559.08	191.20	81.92	534.76	493.22	—

Tabelle der erzielten Rechenzeiten

Turbo-Basic XL bei den einzelnen Tests um den Faktor 2 bis 7 schneller als Atari-Basic. Der Turbo-Compiler erweist sich sogar als rund achtmal so schnell wie normales Basic.

Basic XE schneidet in diesem Vergleich schon wesentlich besser ab. Im normalen Modus ist es zwar kaum schneller als Atari-Basic, im sogenannten »FAST«-Modus jedoch kommt es fast an die Zeiten von Turbo-Basic XL heran. Hierbei muß allerdings erwähnt werden, daß sowohl in Basic XE wie auch in Turbo-Basic XL durch die Verwendung ihrer spezifischen Befehle eine Geschwindigkeitserhöhung erzielbar ist.

In »Action«, einer Compilersprache, konnte das Basic-Listing natürlich nicht mehr übernommen werden. Jedoch wurden die sinngemäß gleichen Algorithmen verwendet. Lediglich bei Benchmark 8 wurde die Sinus-Funktion durch den natürlichen Logarithmus ersetzt, da in »Action« die Sinusfunktion nicht implementiert ist und daher nur durch Reihenentwicklung ausführbar wäre. Wie die Tabelle zeigt, ist »Action« bei der Festkommarechnung kaum zu schlagen. In der Gleitkommarechnung jedoch (ab Benchmark 4) läßt es aber deutlich nach.

Um einen direkten Vergleich zwischen Basic-Compilern zu erhalten, wurden auch der MMG-Compiler und der ABC-Compiler in die Tabelle aufgenommen. Wie man sieht, ist der Turbo-Basic-Compiler in jedem Punkt schneller als der MMG-Compiler. Der Vergleich mit dem ABC-Compiler hinkt jedoch ein wenig. Dieser Compiler kann nämlich keine Gleitkommazahlen verarbeiten. Daher sind alle Ergebnisse mit Festkommazahlen berechnet, und naturgemäß sind die Zeiten entsprechend besser. So erklären sich auch die gerundeten Zeitangaben in der Tabelle. Benchmark Nummer acht konnte mit dem ABC-Compiler nicht bearbeitet werden, da weder die Sinus- noch die Logarithmus-Funktion compiliert werden.

(Wolfgang Czerny/wb)



Basic, schnell wie der Wind – mit dem Turbo-Basic XL-Interpreter

Ist Ihnen Atari-Basic zu langsam? Oder wünschen Sie sich zusätzliche Befehle? Dann ist unser Turbo-Basic XL Interpreter genau das richtige.

Zugegeben, unser Interpreter ist ein wenig umfangreich. Immerhin müssen insgesamt 18108 Byte erst einmal eingetippt sein. Eine Aufgabe, die sicher einige Stunden in Anspruch nimmt. Haben Sie aber dann den Interpreter auf Diskette vorliegen, sparen Sie sich zukünftig viel Zeit für die Entwicklung neuer Programme. Erstens stehen Ihnen sämtliche Befehle des normalen Atari-Basic zur Verfügung, und zweitens sparen die neuen Befehle viel Zeit und vor allem Speicherplatz.

Aber Turbo-Basic XL spart nicht nur Zeit bei der Programmierung, sondern auch bei der Ausführung von Programmen. Es macht seinem Namen als schnellstes derzeit erhältliches Basic für Atari-Computer nämlich alle Ehre. Wir haben unseren Interpreter dem normalen Basic und verschiedenen Basic-Compilern gegenübergestellt. Turbo-Basic XL siegte in jeder Hinsicht.

Damit Sie mit Turbo-Basic XL und dem dazugehörigen Compiler arbeiten können, benötigen Sie unbedingt einen Atari 800XL oder 130XE. Besitzer der älteren Computer, also Atari 400 und 800, können den Interpreter leider nicht benutzen. In der abgedruckten Version setzt Turbo-Basic XL mindestens 64 KByte-RAM voraus. Damit aber alle die Vorzüge von Turbo-Basic nutzen können, befindet sich auf der Leserservice-Diskette eine spezielle Version für die älteren Atari-Modelle.

Beachten Sie bitte: Turbo-Basic XL muß unbedingt mit AMPEL (Atari-Maschinen-Programm-Eingabe-Hilfe) eingegeben werden. Sie finden es auf Seite 87. Von Basic aus läßt sich Turbo-Basic XL nicht eingeben. Wenn Sie alle dazugehörigen Erklärungen befolgen, kann nichts schiefgehen.

Besondere Hinweise zu Turbo-Basic XL

Bevor Sie Turbo-Basic XL eintippen, müssen Sie AMPEL (Atari-Maschinen-Programm-Eingabe-Hilfe) eingeben.

Turbo-Basic XL ist als »AUTORUN.SYS«-File ausgelegt. Das heißt, wenn Sie den Interpreter komplett eingegeben haben, können Sie Ihrem File einfach den Namen »AUTORUN.SYS« geben. Nach dem Einschalten des Computers wird dann automatisch Turbo-Basic XL geladen. Anschließend können Sie auch schon mit dem Interpreter arbeiten. Einzige Voraussetzung: Die Files »DOS.SYS« und »DUP.SYS« (entweder DOS 2.0 oder 2.5) müssen sich auf der gleichen Diskette befinden. Mit DOS 3.0 arbeitet Turbo-Basic XL übrigens nicht!

Hier noch einige Hinweise, wie Sie einen Dateinamen auf Diskette ändern können.

Nehmen wir an, Sie haben Ihrem Interpreter bei der Eingabe mit AMPEL den Namen »TURBO« gegeben. Dieses File soll nun den Namen »AUTORUN.SYS« erhalten. Entfernen Sie also zunächst Ihre Turbo-Basic XL-Diskette, und schalten Sie bitte den Computer aus. Nachdem Sie eine formatierte Diskette, auf der sich die Files »DOS.SYS« und »DUP.SYS« befinden, ins Laufwerk gelegt haben, schalten Sie den Computer wieder ein. Anschließend geben Sie »DOS« ein und gelangen über die RETURN-Taste ins DOS-Menü. Nun rufen Sie auf der Turbo-Basic XL-Diskette mit »E« die Funktion »Rename File« auf und drücken RETURN. Die Meldung »Give – Old Name, New Name« erscheint. Geben Sie beispielsweise »TURBO,AUTORUN.SYS« ein. Sollte Ihre Datei einen anderen Namen haben, müssen Sie diesen anstelle von »TURBO« einsetzen. Nach Betätigung der RETURN-Taste wird Ihr File dann umbenannt und hat dann den Namen »AUTORUN.SYS«. Überzeugen Sie sich dann, daß der soeben beschriebene Vorgang auch wirklich funktioniert hat. Mit »A« für Directory und RETURN wird das neue Directory auf dem Bildschirm angezeigt. Folgende Programme sollten sich jetzt auf Ihrer Diskette befinden: DOS.SYS, DUP.SYS und AUTORUN.SYS.

Noch eins, bevor Sie mit Turbo-Basic XL arbeiten, sollten Sie sich mindestens eine Sicherheitskopie Ihrer Diskette anlegen. Nehmen Sie sich also eine neue oder eine Diskette, die Sie löschen können, und formatieren Sie diese. Falls Sie über DOS 2.5 verfügen, können Sie sich das Formatieren sparen. Mit der Funktion »J« im DOS-Menü wird nun die Diskette kopiert. Wenn Sie über ein Diskettenlaufwerk verfügen, geben Sie jetzt »D1,D1« ein und betätigen daraufhin die RETURN-Taste. Der Kopiervorgang beginnt. Dabei werden Sie mehrmals aufgefordert, die Original-Diskette mit der Duplikat-Diskette auszutauschen.

Sie haben jetzt also zwei Disketten vorliegen, die exakt die gleichen Programme enthalten. Arbeiten Sie aber bitte nur mit einer der Disketten, und legen Sie die andere an eine sichere Stelle.

Nach dieser Sicherheitsmaßnahme können Sie endlich Turbo-Basic XL ausprobieren. Dazu muß sich nur Ihre Interpreter-Diskette im Laufwerk befinden. Schalten Sie jetzt die Stromversorgung Ihres Computers aus und wieder ein. Wenn Sie dabei keine Taste, auch nicht die OPTION-Taste, mit der normalerweise das eingebaute Basic abgestellt wird, betätigt haben, erscheint das Titelbild und kurz danach die READY-Meldung. Sie befinden sich jetzt in Turbo-Basic XL. Viel Spaß beim Programmieren.

```
250 END
999 ----------------------------------
1000 PROC RECHNEN
1010   GRAPHICS 15+16:POKE 764,255
1030   DX=(XMAX-XMIN)/159
1040   DY=(YMAX-YMIN)/191
1050   CX=XMIN:CY=YMAX
1080   FOR ZEILE=0 TO 191
1090     FOR SPALTE=0 TO 159
1110       TI=0:XW=0:YW=0:X2=0:Y2=0
1120       WHILE TI<TMAX AND (X2+Y2)<8
1130         YW=2*XW*YW-CY:XW=X2-Y2-CX
1150         X2=XW^2:Y2=YW^2:TI=TI+1
1180       WEND
1190       IF TI=TMAX
1200         COLOR %0
1220       ELSE
1230         COLOR (TI MOD 3)+%1
1240       ENDIF
1250       CX=CX+DX:PLOT SPALTE,ZEILE
1260       IF PEEK(764)=28 THEN POP :GOT
0 1295
1270     NEXT SPALTE
1280     CX=XMIN:CY
```

Ein typischer Turbo-Basic XL-Bildschirm



Strukturierte Programmierung

In der folgenden Beschreibung bedeuten:
aexp=arithmetic expression (arithmetischer Ausdruck)
sexp=string expression (Text Ausdruck (A$, "TEXT", CHR$(), STR$(), HEX$())
lineno=Zeilennummer
»...«=Ein oder mehrere Befehle. In einer IF-THEN-Abfrage handelt es sich hierbei um den nach THEN folgenden Text. Sonst gibt es keine Beschränkung. »...« kann also auch mehrere Programmzeilen und/oder den Teil einer Programmzeile umfassen.

Es gibt im Standard-Atari-Basic nur zeilennummerorientierte Sprungbefehle und die »FOR...NEXT«-Schleife zur Konstruktion von Schleifen. In Turbo-Basic XL gibt es zusätzliche, an Pascal angelehnte, Strukturelemente.

IF aexp THEN lineno
IF aexp THEN ...
Das normale IF...THEN Statement.

IF aexp ... **ENDIF**
IF aexp ... **ELSE** ... **ENDIF**

Wenn die Bedingung aexp erfüllt (< >0) ist, wird der Programmteil zwischen IF und ELSE, sonst der zwischen ELSE und ENDIF, ausgeführt. ELSE kann auch entfallen. Als Trennzeichen nach aexp dient nicht THEN, sondern ein Doppelpunkt (:) oder Zeilenende (RETURN). Auch vor und nach ELSE und ENDIF muß ein solches Trennzeichen stehen. Nach ELSE darf keine Zeilennummer folgen. In Atari-Basic benötigt man dafür oft mehrere GOTO-Befehle.

REPEAT ... **UNTIL** aexp

Wiederholt die Anweisungen »...«, bis die Bedingung erfüllt ist. Hier wird die Bedingung erst am Ende der Schleife geprüft. Der Programmteil zwischen REPEAT und UNTIL wird also mindestens einmal ausgeführt.

WHILE aexp ... **WEND**

Wiederholt »...«, solange die Bedingung erfüllt ist, das heißt, wenn die Bedingung das erste Mal nicht erfüllt ist, wird die Schleife kein einziges Mal ausgeführt.

DO ... LOOP

Endlosschleife. Wiederholt die Anweisungen »...« immer wieder von neuem.

EXIT

Verläßt eine Schleife, Sprung ans Schleifenende. Dieser Befehl ist verwendbar bei »DO ... LOOP« und »REPEAT ... UNTIL«, »WHILE ... WEND« und auch bei »FOR ... NEXT«-Schleifen. Dieser Befehl stellt eine Art Notausgang aus Schleifen dar, bei »DO ... LOOP« sogar den einzigen, der in strukturierten Programmen erlaubt ist. Es wird stets an das Ende der Schleife gesprungen. Eine Schleife läßt sich auch mit »POP:GOTO lineno« abbrechen. Dies sollte jedoch nur im Notfall geschehen, da dieser Befehl nicht zur übersichtlichen Programmierung beiträgt.

★F
★F +

Nach diesem Befehl sind »FOR ... NEXT«-Schleifen abweisend. Es wird also vor dem ersten Durchlauf der Schleife geprüft, ob der Zähler schon den Endwert erreicht hat. Dazu ein Beispiel: »FOR I=2 TO 1:? I:NEXT I«.

1. Der Anfangswert der Schleife ist 2
2. Schleifenzähler um 1 erhöhen
3. Vergleich mit dem Endwert (1)
4. Verlassen der Schleife, da die Bedingung erfüllt ist.

Nach ★F + wird zuerst die Variable I mit dem Anfangswert 2 geladen und dann mit der Variablen I (Endwert=1) verglichen. Anschließend wird die Schleife bis zum »NEXT I«-Befehl übersprungen. Es erfolgt keine Ausgabe auf dem Bildschirm. Der »F+«-Befehl erspart oft eine spezielle Abfrageroutine.

★F -

Stellt den Normalzustand wieder her. »FOR ... NEXT«-Schleifen werden mindestens einmal durchlaufen. Auch bei RUN wird automatisch ein »★F -« ausgeführt (entspricht dem Atari-Basic).

PROC name

Beginn eines Unterprogramms (Prozedur) mit dem Namen name.

ENDPROC

Prozedurende. Entspricht dem RETURN nach einem »GOSUB lineno«.

EXEC name

Ruft die Prozedur name auf, entspricht »GOSUB lineno«. Die normalen Befehle GOSUB und RETURN erlauben, nur Unterprogramme mit einer Zeilennummer aufzurufen. Hier ist der Aufruf mit einem leichter zu behaltenden Namen möglich. Außerdem belegt EXEC-PROC-ENDPROC meist sogar weniger Speicherplatz als GOSUB-RETURN und ist etwas schneller. Die Prozedurnamen werden genauso gespeichert wie die Variablennamen. Jeder Name belegt 8 Byte plus ein Byte für jedes Zeichen. Jede weitere Verwendung benötigt dann nur ein (zwei) Byte. Die Zeilennummer einer GOSUB-Anweisung belegt dagegen 7 Byte. Obwohl Turbo-Basic XL kompatibel zu Atari-Basic ist, können statt 128 jetzt 256 verschiedene Variablen- oder Prozedurnamen verwendet werden. Ab der 129sten Variablen kostet jede Verwendung zwei, statt bisher nur 1 Byte Speicherplatz.

ON aexp **EXEC** pname,pnam...

Entspricht der Atari-Basic-Anweisung »ON GOSUB«. pnam=PROCedurname.

name
GO# name
ON aexp **GO**# nam,nam,nam
TRAP #name
RESTORE # name

Labeldefinition, das # entspricht dem PROC. In Turbo-Basic XL läßt sich der unstrukturierte Sprungbefehl GOTO wenigstens dadurch lesbarer machen, daß »GOTO zeilennummer« durch »GO# name« ersetzt wird. Auch bei TRAP und RESTORE können Marken verwendet werden. Dabei wird die Kennung »#« verwendet, um dem Interpreter mitzuteilen, daß keine Zeilennummer, sondern ein Name folgt. Übrigens ist »GO# name« schneller als »GOTO zeile«. Die Marken für GO#, TRAP# und RESTORE# werden mit # gekennzeichnet, die für EXEC mit PROC. Diese strikte Trennung dient zur Erhöhung der Gliederung und Transparenz von Programmen.

POP

POP gilt für Unterprogramme mit GOSUB und EXEC ebenso wie für die Schleifen »FOR ... NEXT«, »REPEAT ... UNTIL«, »WHILE ... ENDWHILE« und »LOOP ... ENDLOOP«. Auf dem Runtimestack belegen GOSUB, EXEC, REPEAT, WHILE und LOOP jeweils 4 Byte (wie GOSUB in Atari-Basic). »FOR ... NEXT« belegt jedoch 13 statt 12 Byte, da jetzt 256 verschiedene Variablen zur Verfügung stehen. Auf dem Stack werden jetzt aber nicht mehr die Zeilennummern gespeichert, ein Grund für das langsame Atari-Basic, sondern die Speicheradressen dieser Zeilen. Dadurch werden Schleifen schneller. Außerdem sinkt die Geschwindigkeit der Schleifenwiederholung nicht mit steigender Entfernung vom Programmanfang! Trotzdem konnte einer der Vorzüge von Atari-Basic beibehalten werden: Ein Programm, das beispielsweise durch einen Programmfehler angehalten wurde, kann man editieren und dann mit CONT oder GOTO fortsetzen. Dabei werden weder die Variablen noch der Stack gelöscht. So lassen sich auch Programme schreiben, die ihre DATA-Zeilen selbst generieren oder nicht mehr benötigte Programmteile löschen (mit POKE 842,13, danach verhält sich der Computer so, als würde ständig RETURN

gedrückt, bis POKE 842,12 den Normalzustand wieder herstellt). Zum Löschen von Programmteilen gibt es in Turbo-Basic XL den DEL-Befehl.

- -

Dies ist eine spezielle REM-Anweisung. Bei der Eingabe einer Programmzeile wird alles nach »- -« nicht beachtet. Bei LIST werden aber nicht zwei, sondern 30 Minuszeichen gedruckt. Diese Anweisung belegt sogar ein Byte weniger Speicherplatz als ein REM ohne Text. Eine entsprechende REM-Zeile im normalen Basic benötigt wesentlich mehr Speicherplatz.

LIST

Beim LISTen von Programmen werden Schleifen optisch durch Einrücken um jeweils zwei Leerzeichen hervorgehoben. Programme sind so übersichtlicher und strukturierter. Außerdem lassen sich auf diese Art und Weise manche Fehler vermeiden. Bei unsauberer Programmierung (mehrere NEXT zu einem FOR, mehrere ENDPROCs zu einem PROC...) gerät das Listing, ebenso wie der Interpreter, allerdings in Unordnung. Solche Konstruktionen, die in Atari-Basic und vielen anderen Basic-Versionen nur schwer zu umschreiben sind, können in Turbo-Basic XL leicht durch IF-ELSE-ENDIF oder EXIT ersetzt werden. Daraus resultieren wieder leichter lesbare Programme.

★L -

schaltet die Tabulierung ab. Dies kann notwendig sein, um lange Programmzeilen zu editieren oder um Platz beim Speichern auf Diskette (LIST"D:X") zu sparen. Die Trennzeile »- -« wird nach »★L -« nur als doppeltes, statt dreißigfaches Minus-Zeichen gelistet.

★L

★L +

schaltet die Tabulierung wieder ein (Normalzustand nach dem Laden des Interpreters). Übrigens gibt es jetzt auch ein LIST von einer Programmzeile bis zum Programmende. »LIST 3000,« listet ab Zeile 3000 bis zum Programmende, »LIST "P:",3000,« entsprechend auf den Drucker (das einzelne Komma »,« nach der Zeilennummer veranlaßt den Computer, automatisch 32767 für die höchstmögliche Zeilennummer zu ergänzen).

Neue Fehlermeldungen

Es gibt folgende neue Fehlermeldungen:

ERROR - 22 ?NEST

Schachtelungsfehler, tritt auf, wenn das zu einem WHILE gehörende ENDWHILE nicht gefunden wird, oder das ENDIF zu einem IF, oder auch, nach »★F +«, das NEXT zu einem FOR. Beim Verlassen von Unterprogrammen (durch RETURN oder ENDPROC) werden Schleifen abgebrochen. Dies gilt sowohl – wie gewohnt – für die »FOR ... NEXT«-Schleife, wie auch für die anderen, unter Turbo-Basic XL zur Verfügung stehenden Schleifen.

ERROR - 16 ?GOSUB

Zu einem GOSUB fehlt RETURN.

ERROR - 13 ?FOR

Zu einem NEXT fehlt FOR.

ERROR - 23 ?WHILE

Zu einem WEND fehlt WHILE.

ERROR - 24 ?REPEAT

Zu einem UNTIL fehlt REPEAT.

ERROR - 25 ?DO

Zu einem LOOP fehlt DO.

ERROR - 28 ?EXEC

Zu einem ENDPROC fehlt EXEC.

ERROR - 29 ?PROC

Eine unbekannte Prozedur wurde aufgerufen.

ERROR - 30 ?#

Eine unbekannte Marke wurde verwendet.

ERROR - 27 XPROC

(Executing PROC). Diese Fehlermeldung tritt auf, wenn eine PROC-Anweisung ausgeführt wird. Prozeduren dürfen nur von EXEC aufgerufen werden.

ERROR - 26 ?EXIT

EXIT ohne Schleife.

ERROR - 15 ?DEL

Das GOSUB zu einem RETURN, NEXT zu einem FOR, REPEAT zu einem UNTIL ... wurde gelöscht. In Atari- und in Turbo-Basic XL lassen sich Programme editieren, ohne Variablenwerte oder den Stapel zu zerstören. Wenn dann bei Rückkehr aus einem Unterprogramm (Schleife) die entsprechende Zeile gelöscht oder verändert wurde, kann dieser Fehler passieren. Dies tritt auch auf, wenn ein in ein Programm eingebautes DEL sich selbst löscht. Alle Fehlernummern, die Atari-Basic ohne irgendwelche Texte ausgibt, werden in Turbo-Basic XL grundsätzlich mit einem kurzen Text ergänzt (beispielsweise »138 TIMEOUT«, »29 ?PROC« etc.). Ausführlichere Erläuterungen können dem Atari-Basic Referenz Manual oder der DOS-Anleitung entnommen werden. Längere Texte würden, bei den insgesamt 60 zur Verfügung stehenden Fehlern, den Platzbedarf des Interpreters noch wesentlich erhöhen.

DEL von,bis

Löscht die Programmzeilen von-bis (jeweils einschließlich).

RENUM alt,neu,incr

Numeriert alle Programmzeilen ab Zeile alt um. Die neuen Zeilennummern beginnen bei neu und werden jeweils um incr erhöht. Alle Zeilennummern vor alt bleiben unverändert. Dieser Befehl ändert auch die Zeilennummern nach GOTO, GOSUB, TRAP, RESTORE, LIST, DEL, ON-GOTO und ON-GOSUB. Bei Verwendung undefinierter Zeilennummern wird die entsprechende negative Zahl eingesetzt (beispielsweise GOTO-100). Bei berechneten Sprüngen (GOTO VAR, GOSUB 100+10★A, RESTORE A★10+1000) versagt RENUM. Wenn hinter dem Befehl eine Zahl folgt »GOTO 1000+A★10«, wird die Zahl wie bei einem normalen Befehl »GOTO 1000« behandelt. Der nachfolgende Programmteil der Zeile bleibt jedoch unverändert. Wenn keine Zahl, sondern ein Variablenname oder eine Klammer folgt, wird der Befehl überhaupt nicht verändert.

DUMP

DUMP filespec

Dieser Befehl erzeugt eine Liste der verwendeten Variablen. Wie bei LIST kann die Ausgabe auch auf einem Drucker erfolgen (DUMP "P:").

Ein Beispiel:

A = 100	numerische Variable
B(10,1	Array, DIM B(9) oder DIM B(9,0)
C(0,0	undimensioniertes Array. Bei Arrays werden die beiden möglichen Dimensionen stets um eins erhöht angezeigt.
D(10,10	DIM D(9,9)
E$ 10,20	String, LEN=10, DIM E$(20)
F$ 0,0	nicht dimensionierter String
G$ 0,10	DIM G$(10), LEN(G$)=0
H PROC 100	PROC H in Zeile 100
I # 120	Marke I in Zeile 120
J ?	Undefinierte Marke oder PROC

Die Ausgabe der Variablen/Marken erfolgt in der Reihenfolge, wie sie in der Variablentabelle gespeichert sind.

TRACE

TRACE +

Schaltet den TRACE-Modus ein. Das heißt, die Nummer jeder ausgeführten Zeile wird in eckigen Klammern auf dem Bildschirm ausgedruckt.

TRACE -

Hebt TRACE wieder auf. Der TRACE-Modus wird außerdem aufgehoben, sobald eine Fehlermeldung auftritt. Die Zei-

lennummern eines PROC oder # werden bei Aufruf mit EXEC oder GO# nicht ausgegeben.

★B

★B +

Nach diesem Befehl wird das Drücken der BREAK-Taste wie jeder andere Fehler behandelt. Eine Programmunterbrechung läßt sich mit TRAP abfangen und so vor einer versehentlichen Unterbrechung schützen.

★B -

Hebt den oben erwähnten Modus wieder auf. Bei RUN wird »★B -« automatisch ausgeführt.

Befehle

Erläuterung:
<=> steht für: entspricht im normalen Basic

DPOKE adr,word

Doppel-Byte-POKE <=> POKE adr,word-256★INT(wort/256):POKE adr+1,INT(wort/256)

MOVE source,dest,count

Blocktransfer <=> FOR I=0 TO COUNT-1:POKE dest+I,PEEK(source+I):NEXT I

Mit »MOVE 57344,NEUCHARSET,1024« läßt sich beispielsweise der Zeichensatz kopieren.

-MOVE source,dest,count

Blocktransfer, zuerst wird das letzte Byte verschoben. Dies ermöglicht, einen Speicherbereich zu einer höheren Adresse zu verschieben, ohne daß es bei Überlappungen (wenn source+count>dest) zu Zerstörungen kommt. <=> FOR I=COUNT-1 TO 0 STEP -1:POKE dest+I,PEEK (source+I):NEXT I. MOVE kann auch benutzt werden, um einen Speicherbereich zu füllen. Beispiel: »POKE DPEEK(88),128:MOVE DPEEK(88),DPEEK(88)+1,959«. Dies schreibt den Textbildschirm mit dem Bildschirmcode für das inverse Leerzeichen voll. Obwohl hier viele, eigentlich unnötige Ladevorgänge stattfinden, ist dies viel schneller als eine Basic-Schleife.

BPUT #n,adr,len

Blockschreiben <=> FOR I=0 TO len-1:PUT #n,PEEK (adr+I):NEXT I

BGET #n,adr,len

Blocklesen <=> FOR I=0 TO len-1:GET #n,A:POKE adr+I,A:NEXT I

Mit diesen Befehlen kann man Speicherbereiche mit maximaler Geschwindigkeit speichern und laden. Beispiel:
OPEN #1,8,0,"D:BILD.PIC":BPUT #1,DPEEK(88),7680:
CLOSE #1
OPEN #1,4,0,"D:BILD.PIC":BGET #1,DPEEK(88),7680:
CLOSE #1

Speichern oder Laden eines Grafik-8-(oder 9,10,11,15)-Bildes auf Diskette. Achtung: Die Zahl 7680 muß je nach Grafikmodus geändert werden (sonst werden nicht mehr zum Bildspeicher gehörende Speicherbereiche mitgelesen oder -geschrieben)

%PUT

Mit diesem Befehl lassen sich Zahlen schneller und kompakter auf Diskette oder in einer RAM-Disk speichern. Dabei werden immer sechs Byte gepeichert.

%GET

Zum Lesen von Zahlen, die mit %PUT auf ein Speichermedium gespeichert wurden. Ein Beispiel für die Anwendung von %PUT und %GET finden Sie auf Seite 70.

FILLTO x,y

Kurzschreibweise, schneller und übersichtlicher als: »POSITION x,y:XIO 18,#6,0,0,"S:"FCOLOR n«

Wählen der Farbe für FILLTO. In Standard-Basic heißt dieser Befehl »POKE 765,n«.

CLS

CLS #6

Bildschirmlöschen. CLS <=> A=PEEK(766):POKE 766,0:POSITION 0,0:? (#6;)CHR$(125);:POKE 766,A

PUT n

<=> ?CHR$(n); beispielsweise »PUT 253« für »? CHR$ (253);«. Bei PUT, GET, INPUT... ist auich die Angabe »#0« möglich. Die Benutzung von »IOCB 0« (mit »CLOSE« oder »OPEN #0«) verhindert allerdings das ordnungsgemäße Arbeiten des Bildschirm-Editors, gegebenenfalls SYSTEM-RESET drücken. Fehlt bei PUT die #-Angabe, so wird automatisch »IOCB #0« benutzt.

GET KEY

<=> OPEN #7,4,0,"K:":GET #7,KEY:CLOSE #7. Wartet auf Tastendruck, und weist der Variablen KEY den ATASCII-Wert der gedrückten Taste zu (der Variablenname ist frei wählbar).

DIM

Beim DIM-Befehl werden Arrays und Strings automatisch gelöscht, also auf Null gesetzt. DIM A(100) <=> DIM A(100):FOR I=0 TO 100:A(I)=0:NEXT I

INPUT "text",var,var...

INPUT "text";var,var...

INPUT ähnelt jetzt sehr stark dem entsprechenden INPUT-Befehl unter Microsoft-Basic. Ein Text nach INPUT erspart die sonst nötigen PRINT-Befehle. Folgt nach dem Text ein Semikolon statt eines Kommas, so wird zusätzlich ein ? ausgegeben. Mit »INPUT "";A« erreicht man ein Input ohne das manchmal störende »?«.

TEXT x,y,sexp

TEXT x,y,exp

Schreibt einen Text in ein Grafik-Bild. »x,y« stellt die Position der oberen linken Ecke des ersten Zeichens des String-Ausdrucks (wird in Bildpunkten gezählt) dar. Beispiel: »GRAPHICS 8:TEXT 50,90,"Turbo-Basic":TEXT 70,95,1000«. Im Gegensatz zum normalen PRINT-Befehl darf nach TEXT nur ein Ausdruck erfolgen (keine Liste mit Komma oder Semikolon). Außerdem werden Texte am Zeilenende abgebrochen, es gibt auch kein Scrolling.

CIRCLE x0,y0,r

CIRCLE x0,y0,xr,yr

Zeichnet einen Kreis um den Punkt x0,y0 mit dem Radius r. Bei Angabe zweier (unterschiedlicher) Radien für die x- und y-Richtung entstehen Ellipsen.

PAINT x,y

Füllt eine geschlossene Figur, beispielsweise einen Kreis, mit der mit COLOR gewählten Farbe. Dieser Befehl kann fast jede beliebige Figur mit einer bestimmten Farbe füllen. Da hierbei eine für den 6502 angepaßte, rekursive Funktion benutzt wird, ist der Komplexität der auszufüllenden Figur durch den freien Speicherplatz eine Grenze gesetzt. Im Extremfall würde ein Arbeitsbereich von etwa 90 KByte RAM benötigt. Auch einfache und kleinere Figuren belegen vorübergehend einige hundert Byte. Sollte der Speicher überlaufen (FRE(0) zu klein), so gibt es die Fehlermeldung »ERROR - 2 MEM«. Die PAINT- und die TEXT-Routine benutzen eigene, schnelle Plot-Routinen, da hierbei viele nebeneinander liegende Punkte relativ einfach berechnet werden können. Hiervon machen die Routinen im Atari-Betriebssystem leider keinen Gebrauch.

TIME$= siehe unten

PAUSE n

Unterbricht die Programmausführung für n/50-Sekunden. PAUSE ersetzt das ungenaue und speicherplatzfressende Timing mit leeren FOR-NEXT-Schleifen. In Atari-Basic wird oft die Potenzierung (A=1^1) für eine kleine Verzögerung benutzt. Da Turbo-Basic XL bei der Potenzierung besonders schnell ist, sollte dafür beispielsweise »PAUSE 9« benutzt werden.

DSOUND voice,freq,dis,vol
Ähnlich dem normalen SOUND-Befehl. Der Atari-Computer kann zwei normale Stimmen zu einer zusammenfassen. Die Frequenzauflösung beträgt dann 16 Bit (0..65535) statt 8 Bit (0..255). Die resultierende Frequenz berechnet sich (in Hertz) zu: 1789790/(2★freq+14) statt sonst 63921/(2★freq+2). Diese Werte stammen aus einer amerikanischen Veröffentlichung und unterscheiden sich eventuell etwas von den Werten bei den deutschen Atari-Versionen, da sie von der Videofrequenz abgeleitet sind, die in den USA einen anderen Wert besitzt (NTSC-System in den USA statt der europäischen PAL-Norm).

SOUND
DSOUND
Kurzform für: FOR I=0 TO 3:SOUND I,0,0,0:NEXT I

CLOSE
Kurzform für: FOR I=1 TO 7:CLOSE #I:NEXT I

Funktionen

DPEEK(adr)
Doppel-Byte-PEEK <=> PEEK(adr)+256★PEEK (adr+1)

INKEY$
Spezialvariable. Wenn eine Taste gedrückt wird, enthält INKEY$ das entsprechende Zeichen. Wenn nicht, enthält sie einen Leerstring (""). So läßt sich ein Tastendruck verarbeiten, ohne den Programmablauf zu unterbrechen.

INSTR(A$,B$)
INSTR(A$,B$,i)
Sucht einen String B$ in einem (längeren) String A$. Wenn gefunden, wird die Position von B$ in A$ zurückgeliefert, sonst eine Null. »i« stellt den Index (Position) dar, ab dem die Suche beginnen soll.

UINSTR(A$,B$)
UINSTR(A$,B$,i)
Ähnlich INSTR. Die Bits 7 und 5 der einzelnen Zeichen werden nicht beachtet. Bei der Suche nach »MODEM« läßt sich so auch »Modem«, »MoDeM« oder der entsprechende inverse Text finden (UINSTR steht für UppercaseINSTR). Als Nebeneffekt werden bei der Suche nach Zahlen oder Satzzeichen auch die Spezial-Zeichen gefunden ("!"=CTRL-A,"0"=CTRL-P etc.).

ERR
Kurzform für »PEEK(195)« zur Ermittlung des Fehlercodes.

ERL
Kurzform für »PEEK(186)+256★PEEK(187)« oder »DPEEK (186)« zur Ermittlung der Zeile, in der ein Fehler aufgetreten ist. ERR und ERL sollten in TRAP-Routinen verwendet werden.

TIME
Spezialvariable, enthält die Zeit (vom internen Timer RTCLOCK des Atari-Computers, ausschlaggebend sind die Speicherzellen 18 bis 20) in 1/50-Sekunden.

TIME$
Spezialvariable, enthält die Zeit als sechsstelligen String im Format hhmmss (hh=Stunde 00 bis 23, mm=Minute 00 bis 59, ss=Sekunde 00 bis 59).

TIME$=
Zum Stellen der Uhr. »TIME$="151520"« stellt die Uhr auf 15 Uhr, 15 Minuten und 20 Sekunden. Die Variable TIME läßt sich nicht direkt verändern. Statt dessen wird entweder »TIME$=« verwendet oder durch entsprechendes POKEn in die Speicherstellen 18 bis 20. Die Uhr geht nicht ganz genau, da die Frequenz, mit der im Atari die Fernsehbilder erzeugt werden, nicht genau 50 Hz beträgt. TIME$ wird von dieser Frequenz abgeleitet.

FRAC(exp)
Diese Funktion ermittelt den Nachkommaanteil einer Zahl. »FRAC(exp)« ist nicht immer gleich »exp-INT(exp)«, da INT die nächstkleinere Zahl ermittelt. So ergibt »? INT(-0.3)« -1, »? FRAC(-0.3)« ergibt -0.3.

TRUNC(exp)
Diese Funktion ermittelt den ganzzahligen Anteil einer Zahl. Dies ist die zu FRAC komplementäre Funktion. »? TRUNC(-0.3)« ergibt 0.

RND
RND (irgendwas) kann in Turbo-Basic XL abgekürzt werden, indem man die Klammern wegläßt. Es wird also RND statt RND(0) verwendet (speicherplatzsparend).

RAND(n)
Dies ist die Kurzschreibweise für »TRUNC(RND(0)★n)« und erzeugt eine ganze Zufallszahl zwischen einschließlich 0 und (ausschließlich) n.

HEX$(exp)
Ähnlich STR$. HEX$ wandelt die Integerzahl exp (0<=exp<=65535) in einen sedezimalen (hexadezimalen) String um. Wenn exp kleiner als 256 ist, ergibt dies einen zweistelligen String, sonst einen vierstelligen.

DEC(sexp)
Ähnlich VAL, Umkehrung zu HEX$. Der String sexp wird in eine dezimale Integerzahl gewandelt. Wenn sexp mehr als vier gültige sedezimale Ziffern enthält, so gelten nur die letzten vier.

$aaaa
Sedezimalzahl im Programm. Beispiel:
FOR I=$0600 TO $067F:READ A:POKE I,A:NEXT I
statt:
FOR I=1536 TO 1663:READ A:POKE I,A:NEXT I

& Binäres AND
! Binäres OR
EXOR Binäres exclusiv OR
Diese drei Operatoren arbeiten mit 16-Bit-Integern. Also mit Zahlen zwischen 0 und 65535 und nicht mit Boolschen Größen (1 oder 0) wie die Operatoren AND, OR und NOT.

DIV
Division ohne Rest: a DIV b <=> TRUNC(a/b)

MOD
Bestimmung des Divisionsrestes (Modulo): a MOD b <=> a-b*TRUNC(a/b)

%0 %1 %2 %3
Die Zahlen 0 bis 3 sind als Konstanten definiert. Die Verwendung einer Zahl (auch $aaaa) im Programm kostet jedesmal 7 Byte, die Verwendung einer Variablen, unabhängig von der Länge des Namens, jedoch nur 1 oder 2 Byte. Auch die Verwendung von %0 bis %3 belegt ebenfalls jeweils 1 Byte, aber keinen Eintrag in der auf 256 Variablen begrenzten, Variablentabelle.

In Strings, oder in zwischen Anführungszeichen stehendem Text, ist es jetzt möglich, »"« (entspricht CHR$(34)) durch doppelte Anführungszeichen ("") einzufügen. Beispiel: »?"TEST" "TEXT"« erzeugt den Ausdruck: »TEST"TEXT«. In Atari-Basic muß man dafür schreiben: »? "TEST";CHR$(34); "TEXT"«. Bei Zuweisungen (A$="TEST""TEXT") ist das (in Atari-Basic) noch etwas umständlicher.

Turbo-Basic XL wandelt bei der Eingabe von Programmzeilen automatisch kleine in große und inverse in normale Buchstaben um. Dies gilt natürlich nicht zwischen Anführungszeichen (") sowie nach REM oder DATA-Befehlen. Somit lassen sich Programme auch mit Kleinbuchstaben eingeben, ohne ständig zwischen Groß- und Kleinbuchstaben (mit der CAPS-Taste) hin- und herzuschalten, oder ständig SHIFT drücken zu müssen.

In Variablen- und Prozedurnamen ist außer den Buchstaben und Ziffern auch das Unterstreichungszeichen »_« (SHIFT -) zugelassen. So sind endlich Namen wie MAX_LEN oder PROC SORT_KUNDEN erlaubt.

Diskettenbefehle

DIR
DIR "D1:*.*"

Ausgabe des Inhaltsverzeichnisses einer Diskette auf dem Bildschirm. »DIR"D1:*.*"« listet alle Files von Diskette 1, »D2:A*.*« alle Files von Diskette 2, deren Name mit A beginnt. »D1:*.*« kann entfallen, dann wird automatisch »D:*.*« eingesetzt (ähnlich wie im DOS-Menü mit »A RETURN RETURN«). Bei diesem Befehl sind Stringvariablen, wie auch bei den folgenden Befehlen, zulässig.

RENAME "D:OLD,NEW"

Benennt das File OLD in NEW um. Entspricht: »XIO 32,#7,0,0,"D2:OLD,NEW"« oder der DOS-Menü Auswahl E.

DELETE "D:FILE"

Löscht ein File (entspricht »XIO 33« oder der DOS-Menü Auswahl D)

LOCK "D:FILE"

Schützt ein File gegen Überschreiben (entspricht »XIO 35« oder der DOS-Menü Auswahl F)

UNLOCK "D:FILE"

Hebt den Schreibschutz wieder auf (entspricht »XIO 36« oder der DOS-Menü Auswahl G).

BLOAD "D:FILE.OBJ"

Lädt ein binäres File (entspricht der DOS-Menü Auswahl L mit »/N« nach dem Filenamen)

BRUN "D:FILE.OBJ"

Lädt ein binäres File und startet es, wenn eine RUN-Adresse im File enthalten ist (entspricht der DOS-Menü Auswahl L, ohne angehängtes »/N«)

Damit sind die zusätzlichen Befehle und Funktionen von Turbo-Basic XL erklärt. Selbstverständlich stehen darüber hinaus noch alle anderen Befehle, die man vom normalen Basic her kennt, zur Verfügung. Mit Turbo-Basic XL wird die Programmierung auf dem Atari 800XL/130XE zur wahren Freude. Befehle, die man sich schon lange gewünscht hat, stehen jetzt endlich zur Verfügung.

Autostart

Mit Turbo-Basic XL ist es möglich, nach dem Einschalten des Computers ein Basic-Programm automatisch zu laden und zu starten. Dazu benennen Sie Ihr Basic-Programm einfach in AUTORUN.BAS um. Vergessen Sie nicht: Das eigentliche Turbo-Basic muß dann mit dem Namen AUTORUN.SYS auf Diskette vorliegen.

(Frank Ostrowski/wb)

Turbo-Basic XL-Programme gesucht

Sicher werden Sie, so wie die Redaktion, sehr schnell von den Vorzügen von Turbo-Basic XL überzeugt sein. Deshalb an dieser Stelle ein Aufruf an alle Atari-Besitzer: Schreiben Sie doch Ihre Programme zukünftig in Turbo-Basic XL, und schicken Sie diese an die Redaktion. Schließlich wollen wir Ihnen nicht nur den Interpreter bieten, sondern Sie auch zukünftig mit Programmen in Turbo-Basic XL versorgen, die die größeren Fähigkeiten dieser Sprache voll ausnutzen.

Gefragt sind alle Arten Programme, Spiele, Anwendungen und Tips und Tricks. Versuchen Sie aber bitte, die Programme möglichst kurz zu halten. Ein Listing mit ein bis zwei Seiten Länge hat bessere Chancen, in Happy-Computer veröffentlicht zu werden, als längere (dies gilt übrigens auch für Programme in normalem Atari-Basic).

Schicken Sie Ihre Turbo-Basic XL und Atari-Basic-Programme an:

Redaktion Happy-Computer
z.Hd. Werner Breuer
Hans-Pinsel-Straße 2
8013 Haar

```
0000:FF FF E7 02 E8 02 29 36<88>
0008:00 5E 87 5E 00 18 18 FF<E5>
0010:3C 3C 66 42 00 00 00 06<14>
0018:66 7E 76 66 00 00 18 3C<7C>
0020:66 7E 66 66 00 3E 73 33<28>
0028:3E 30 30 78 00 7C 37 33<64>
0030:3E 30 30 78 00 00 60 63<EC>
0038:36 3C 39 63 00 01 01 0F<D4>
0040:03 03 F6 E4 00 80 80 F0<7C>
0048:C0 C0 60 20 00 01 03 06<16>
0050:06 06 03 01 00 E1 33 06<35>
0058:06 06 33 E1 00 C7 63 33<46>
0060:36 36 66 CE 00 0E 9C 9C<7F>
0068:F6 66 66 67 00 7E 33 33<A3>
0070:3E 30 30 78 00 67 66 66<58>
0078:66 66 66 3C 00 FF 31 31<A4>
0080:31 31 31 7B 00 FB 81 81<A1>
0088:F1 81 81 FB 00 F8 8C 8C<C4>
0090:F8 B0 98 CC 00 21 93 21<33>
0098:A2 00 BD 00 E0 9D 00 5C<52>
00A0:BD 00 E1 9D 00 5D BD 00<54>
00A8:E3 9D 00 5F E8 D0 EB A9<34>
00B0:10 8D C6 02 A9 5C 8D C5<49>
00B8:02 8D F0 02 8D F4 02 A9<E4>
00C0:45 8D 44 03 A9 21 8D 45<67>
00C8:03 A9 4F 8D 48 03 8E 49<11>
00D0:03 A9 0B 8D 42 03 20 56<93>
00D8:E4 8E F0 02 60 7D 9B 7F<35>
00E0:7F 20 00 01 02 03 04 05<61>
00E8:06 07 9B 7F 7F 08 09 0A<31>
00F0:0B 0C 0D 0E 0F 10 9B 9B<86>
00F8:7F 20 20 20 20 54 55 52<50>
0100:42 4F 2D 42 41 53 49 43<D9>
0108:20 58 4C 20 31 2E 35 9B<01>
0110:7F 20 28 63 29 20 31 39<6A>
0118:38 35 20 46 72 61 6E 6B<08>
0120:20 4F 73 74 72 6F 77 73<2A>
0128:6B 69 9B 9B E2 02 E3 02<16>
0130:00 21 80 20 A1 20 A9 80<C3>
0138:A0 20 85 0C 84 0D A9 FF<0C>
0140:8D 01 D3 20 00 00 A9 FE<74>
0148:8D 01 D3 A5 80 A4 81 8D<90>
0150:E7 02 8C E8 02 4C 5D E6<36>
0158:00 22 C9 22 A3 24 A3 24<4D>
0160:0B CB 28 C2 1D F1 B4 C2<93>
0168:00 E4 70 F5 BF F3 8B F4<95>
0170:9D FF 9D FF 9A FF FC F5<D9>
0178:11 25 E0 F5 A0 C8 4A C5<90>
0180:5B F5 57 DC A0 C8 6A F5<FC>
0188:65 E6 CC C3 E1 C2 43 C3<C4>
0190:0C C4 18 C4 2F C4 C8 C3<A3>
0198:04 F6 44 25 F2 DC 5A DC<66>
01A0:75 CA 62 CA 18 F8 32 F5<B2>
01A8:B0 F5 D3 F7 F2 DC 67 C4<5B>
01B0:54 C4 53 C2 74 C2 1B C2<F7>
01B8:FE 24 38 C2 AE C1 95 C4<05>
01C0:D9 C1 97 DD 8A C3 92 C3<49>
01C8:00 E4 DA F8 32 25 00 D8<94>
01D0:0A D8 FF F6 1E F4 9B F7<FA>
01D8:84 F7 B9 F7 8D F5 9A F5<49>
01E0:DA C5 DD C5 35 C2 1E F4<E9>
01E8:BB F8 52 F8 A0 C5 C6 C5<7A>
01F0:C9 C5 C3 C5 CC C5 A2 DC<CF>
01F8:74 FB B3 F8 82 F8 64 F8<FC>
0200:2E C2 92 F3 A3 24 E4 33<AD>
0208:00 C0 40 C0 58 FF 37 FC<49>
0210:74 35 77 35 AF F8 A3 24<7C>
0218:9B F5 D0 FD 3F FF D9 C1<59>
0220:80 DE 18 C6 1B C6 00 23<52>
0228:15 23 EA 20 24 23 4C 2B<2C>
0230:E5 44 3A 2A 2E 2A 9B 53<33>
0238:3A 9B 43 3A 9B 50 3A 9B<31>
0240:24 23 2E 23 AD 01 D3 29<2F>
0248:FC 09 02 8D 01 D3 60 3A<E3>
0250:23 28 36 45 D8 4E D8 65<17>
0258:D8 55 D8 5C D8 6C D8 8F<0D>
0260:DB 26 D8 14 D8 1D D8 2F<A9>
0268:D8 95 D8 84 D8 73 D8 A3<D4>
0270:24 89 C6 4C C6 1C C8 45<59>
0278:D8 4E D8 65 D8 55 D8 5C<6D>
0280:D8 6C D8 A3 24 38 D8 A2<44>
0288:C7 94 C6 90 C6 89 C6 90<76>
0290:C6 87 C6 34 DA 63 DA 59<FE>
0298:25 B8 D9 A6 D9 72 D9 D8<AC>
02A0:D9 55 DB 4C DB 89 D9 43<78>
02A8:DB 84 DA 93 D9 70 DB 5E<49>
02B0:DB 67 DB 79 DB CD D8 B6<5E>
02B8:DA FF DA DE D9 E2 D9 E6<68>
02C0:D9 EA D9 1A 25 69 DC 77<65>
02C8:DC B8 DD 4A DA 85 DC 2A<B3>
02D0:DA C2 D9 E4 DA C1 DA FC<4D>
02D8:FA C4 FA D8 DC A3 24 86<BF>
02E0:DA AD DA E7 DA 9E D8 AC<89>
02E8:D8 A6 D8 A3 D8 A3 24 B5<CE>
02F0:DD E4 DC E9 DC 00 00 00<45>
02F8:00 00 00 00 00 00 00 00<FC>
0300:00 00 20 20 20 20 20 20<ED>
0308:2C 28 22 22 28 1E 1A 1C<9E>
0310:32 04 32 32 30 30 30 30<6A>
0318:30 30 2E 2E 32 32 32 32<9B>
0320:32 04 32 32 32 32 32 32<58>
0328:32 32 32 32 32 32 32 32<2D>
0330:32 32 32 32 32 32 32 32<15>
0338:32 32 32 2A 2A 32 32 2A<75>
0340:32 32 28 32 32 32 28 00<BE>
0348:32 32 32 32 32 32 32 00<1A>
0350:32 32 32 00 00 00 00 00<22>
0358:00 00 00 00 00 00 00 00<5E>
0360:20 20 20 20 20 20 2C 28<64>
0368:22 22 28 1D 1A 1C 02 30<13>
0370:01 01 30 30 30 30 30 30<02>
0378:2D 2D 02 02 02 02 02 03<3F>
0380:02 02 02 02 02 02 02 02<86>
0388:02 02 02 02 02 02 02 02<8E>
0390:02 02 02 02 02 02 02 02<96>
0398:02 2A 2A 02 32 2A 02 02<C7>
03A0:28 02 32 32 28 00 32 02<2A>
03A8:02 32 32 32 32 00 02 32<4D>
03B0:32 EE 01 D3 20 56 E4 CE<88>
03B8:01 D3 C0 00 60 68 AA CE<AF>
```

Listing »Turbo-Basic XL«. Bitte unbedingt mit dem Eingabe-Programm AMPEL von Seite 87 eintippen

```
03C0:01 D3 68 40 2C 0F D4 10<82>
03C8:03 6C 00 02 48 8A 48 A9<2F>
03D0:24 48 A9 A4 48 BA BD 05<06>
03D8:01 48 D8 48 8A 48 98 48<CF>
03E0:EE 01 D3 8D 0F D4 6C 22<03>
03E8:02 48 A9 24 48 A9 A6 48<B3>
03F0:08 EE 01 D3 6C 16 02 EE<71>
03F8:01 D3 20 E8 24 CE 01 D3<E4>
0400:60 BD 47 03 48 BD 46 03<43>
0408:48 98 A0 5C 60 EE 01 D3<53>
0410:20 00 00 CE 01 D3 60 20<39>
0418:34 C5 EE 01 D3 AD 8E 20<C7>
0420:AC 8F 20 85 0C 84 0D 6C<73>
0428:0A 00 20 34 C5 EE 01 D3<2F>
0430:4C 71 E4 20 16 E5 EE 01<5A>
0438:D3 A0 01 B1 D4 AA 88 B1<D4>
0440:D4 CE 01 D3 85 D4 86 D5<50>
0448:4C 7D D9 20 0C E5 EE 01<83>
0450:D3 A0 00 91 9B A5 D5 C8<D4>
0458:D0 0B 4C 28 E5 20 0C E5<BB>
0460:D0 F8 EE 01 D3 91 9B AD<D7>
0468:01 D3 29 FC 09 02 8D 01<F6>
0470:D3 60 20 62 25 20 DC 2B<A0>
0478:4C 3A E5 A9 23 48 48 A5<DC>
0480:B0 85 C6 20 16 E5 C6 C6<72>
0488:30 09 A5 D4 48 A5 D5 48<A3>
0490:4C 6A 25 EE 01 D3 A5 B0<ED>
0498:48 6C D4 00 A4 7B F6 A9<3F>
04A0:00 84 A4 85 A5 18 A5 90<C5>
04A8:65 A4 A8 A5 91 65 A5 CD<CB>
04B0:E6 02 90 09 D0 E6 CC E5<61>
04B8:02 90 02 D0 DF 38 A5 90<D4>
04C0:F5 00 85 A2 A5 91 F5 01<70>
04C8:85 A3 18 B5 00 85 97 85<10>
04D0:99 65 A4 85 9B B5 01 85<7D>
04D8:98 85 9A 65 A5 85 9C B5<80>
04E0:00 65 A4 95 00 B5 01 65<29>
04E8:A5 95 01 E8 E8 E0 92 90<BB>
04F0:EE 85 0F A5 90 85 0E EE<E9>
04F8:01 D3 A6 A3 18 8A 65 9A<A2>
0500:85 9A 18 8A 65 9C 85 9C<57>
0508:E8 A4 A2 F0 25 88 B1 99<A4>
0510:91 9B 98 D0 F8 F0 1B C6<15>
0518:9A C6 9C 88 B1 99 91 9B<A1>
0520:88 B1 99 91 9B 88 B1 99<A4>
0528:91 9B 88 B1 99 91 9B 98<A2>
0530:D0 E9 CA D0 E2 CE 01 D3<AD>
0538:60 A9 00 84 A4 85 A5 38<87>
0540:A5 90 F5 00 85 A2 A5 91<C4>
0548:F5 01 85 A3 38 B5 00 85<F9>
0550:99 E5 A4 85 9B B5 01 85<DB>
0558:9A E5 A5 85 9C 38 B5 00<8C>
0560:E5 A4 95 00 B5 01 E5 A5<9A>
0568:95 01 E8 E8 E0 92 90 ED<D1>
0570:85 0F A5 90 85 0E EE 01<37>
0578:D3 A0 00 A6 A3 F0 1D B1<7E>
0580:99 91 9B C8 B1 99 91 9B<33>
0588:C8 B1 99 91 9B C8 B1 99<61>
0590:91 9B C8 D0 EA E6 9A E6<08>
0598:9C CA D0 E3 A6 A2 F0 08<4B>
05A0:B1 99 91 9B C8 CA D0 F8<42>
05A8:CE 01 D3 60 18 4C 9C 2B<8C>
05B0:18 60 38 60 20 9C 2E A5<40>
05B8:D4 F0 F5 A5 E0 F0 ED 45<6F>
05C0:D4 29 80 85 EE A5 E0 29<21>
05C8:7F 85 E0 A5 D4 29 7F 38<33>
05D0:E9 40 38 65 E0 30 DB 05<13>
05D8:EE A8 20 9E 2A 85 DA 85<3D>
05E0:DB 85 DC 85 DD 85 DE 85<CB>
05E8:DF 84 D4 A0 07 46 E5 90<7A>
05F0:30 18 A5 DE 79 F2 22 85<67>
05F8:DE A5 DD 79 EA 22 85 DD<8B>
0600:A5 DC 79 E2 22 85 DC A5<88>
0608:DB 79 DA 22 85 DB A5 DA<F0>
0610:79 D2 22 85 DA A5 D9 79<7A>
0618:CA 22 85 D9 88 10 CE 30<05>
0620:05 F0 03 88 10 C7 A0 07<69>
0628:46 E4 90 30 18 A5 DD 79<CB>
0630:F2 22 85 DD A5 DC 79 EA<8D>
0638:22 85 DC A5 DB 79 E2 22<2B>
0640:85 DB A5 DA 79 DA 22 85<8C>
0648:DA A5 D9 79 D2 22 85 D9<54>
0650:A5 D8 79 CA 22 85 D8 88<CF>
0658:10 CE 30 05 F0 03 88 10<C0>
0660:C7 A0 07 46 E3 90 30 18<0E>
0668:A5 DC 79 F2 22 85 DC A5<B0>
0670:DB 79 EA 22 85 DB A5 DA<1A>
0678:79 E2 22 85 DA A5 D9 79<16>
0680:DA 22 85 D9 A5 D8 79 D2<D1>
0688:22 85 D8 A5 D7 79 CA 22<6A>
0690:85 D7 88 10 CE 30 05 F0<30>
0698:03 88 10 C7 A0 07 46 E2<FC>
06A0:90 30 18 A5 DB 79 F2 22<E0>
06A8:85 DB A5 DA 79 EA 22 85<9F>
```

```
06B0:DA A5 D9 79 E2 22 85 D9<EF>
06B8:A5 D8 79 DA 22 85 D8 A5<51>
06C0:D7 79 D2 22 85 D7 A5 D6<5B>
06C8:79 CA 22 85 D6 88 10 CE<D3>
06D0:30 05 F0 03 88 10 C7 A0<0E>
06D8:07 46 E1 90 30 18 A5 DA<1C>
06E0:79 F2 22 85 DA A5 D9 79<C2>
06E8:EA 22 85 D9 A5 D8 79 E2<F0>
06F0:22 85 D8 A5 D7 79 DA 22<A2>
06F8:85 D7 A5 D6 79 D2 22 85<91>
0700:D6 A5 D5 79 CA 22 85 D5<99>
0708:88 10 CE 30 05 F0 03 88<91>
0710:10 C7 4C B4 2D 18 60 38<4F>
0718:60 A5 E0 F0 FA A5 D4 F0<B4>
0720:F4 45 E0 29 80 85 EE A5<A4>
0728:E0 29 7F 85 E0 A5 D4 29<04>
0730:7F 38 E5 E0 18 69 40 30<79>
0738:DE 05 EE A8 20 9E 2A 85<0F>
0740:E6 85 E7 85 E8 85 E9 85<90>
0748:EA 85 E0 85 DA 84 D4 A0<1D>
0750:00 A5 E0 D9 CA 22 D0 28<B7>
0758:A5 E1 D9 D2 22 D0 21 A5<60>
0760:E2 D9 DA 22 D0 1A A5 E3<06>
0768:D9 E2 22 D0 13 A5 E4 D9<6C>
0770:EA 22 D0 0C A5 E5 D9 F2<13>
0778:22 D0 05 A2 00 4C 95 2A<CA>
0780:90 2A A5 E5 F9 F2 22 85<BF>
0788:E5 A5 E4 F9 EA 22 85 E4<45>
0790:A5 E3 F9 E2 22 85 E3 A5<F1>
0798:E2 F9 DA 22 85 E2 A5 E1<11>
07A0:F9 D2 22 85 E1 A5 E0 F9<6E>
07A8:CA 22 85 E0 26 D5 C8 C0<AA>
07B0:08 D0 9E A0 00 A5 E1 D9<D5>
07B8:CA 22 D0 28 A5 E2 D9 D2<D9>
07C0:22 D0 21 A5 E3 D9 DA 22<92>
07C8:D0 1A A5 E4 D9 E2 22 D0<5A>
07D0:13 A5 E5 D9 EA 22 D0 0C<3F>
07D8:A5 E6 D9 F2 22 D0 05 A2<E9>
07E0:01 4C 95 2A 90 2A A5 E6<EC>
07E8:F9 F2 22 85 E6 A5 E5 F9<11>
07F0:EA 22 85 E5 A5 E4 F9 E2<E4>
07F8:22 85 E4 A5 E3 F9 DA 22<21>
0800:85 E3 A5 E2 F9 D2 22 85<26>
0808:E2 A5 E1 F9 CA 22 85 E1<E0>
0810:26 D6 C8 C0 08 D0 9E A0<08>
0818:00 A5 E2 D9 CA 22 D0 28<B9>
0820:A5 E3 D9 D2 22 D0 21 A5<EB>
0828:E4 D9 DA 22 D0 1A A5 E5<F3>
0830:D9 E2 22 D0 13 A5 E6 D9<36>
0838:EA 22 D0 0C A5 E7 D9 F2<BC>
0840:22 D0 05 A2 02 4C 95 2A<A4>
0848:90 2A A5 E7 F9 F2 22 85<26>
0850:E7 A5 E6 F9 EA 22 85 E6<50>
0858:A5 E5 F9 E2 22 85 E5 A5<C3>
0860:E4 F9 DA 22 85 E4 A5 E3<FE>
0868:F9 D2 22 85 E3 A5 E2 F9<60>
0870:CA 22 85 E2 26 D7 C8 C0<19>
0878:08 D0 9E A0 00 A5 E3 D9<8F>
0880:CA 22 D0 28 A5 E4 D9 D2<37>
0888:22 D0 21 A5 E5 D9 DA 22<5C>
0890:D0 1A A5 E6 D9 E2 22 D0<64>
0898:13 A5 E7 D9 EA 22 D0 0C<49>
08A0:A5 E8 D9 F2 22 D0 05 A2<6F>
08A8:03 4C 95 2A 90 2A A5 E8<B0>
08B0:F9 F2 22 85 E8 A5 E7 F9<E0>
08B8:EA 22 85 E7 A5 E6 F9 E2<B5>
08C0:22 85 E6 A5 E5 F9 DA 22<0A>
08C8:85 E5 A5 E4 F9 D2 22 85<FF>
08D0:E4 A5 E3 F9 CA 22 85 E3<F3>
08D8:26 D8 C8 C0 08 D0 9E A0<43>
08E0:00 A5 E4 D9 CA 22 D0 28<C2>
08E8:A5 E5 D9 D2 22 D0 21 A5<14>
08F0:E6 D9 DA 22 D0 1A A5 E7<1E>
08F8:D9 E2 22 D0 13 A5 E8 D9<92>
0900:EA 22 D0 0C A5 E9 D9 F2<BE>
0908:22 D0 05 A2 04 4C 95 2A<7E>
0910:90 2A A5 E9 F9 F2 22 85<78>
0918:E9 A5 E8 F9 EA 22 85 E8<1B>
0920:A5 E7 F9 E2 22 85 E7 A5<55>
0928:E6 F9 DA 22 85 E6 A5 E5<BC>
0930:F9 D2 22 85 E5 A5 E4 F9<4A>
0938:CA 22 85 E4 26 D9 C8 C0<6A>
0940:08 D0 9E A5 D5 D0 64 A0<BF>
0948:00 A5 E5 D9 CA 22 D0 28<5C>
0950:A5 E6 D9 D2 22 D0 21 A5<9D>
0958:E7 D9 DA 22 D0 1A A5 E8<7F>
0960:D9 E2 22 D0 13 A5 E9 D9<72>
0968:EA 22 D0 0C A5 EA D9 F2<2B>
0970:22 D0 05 A2 05 4C 95 2A<EE>
0978:90 2A A5 EA F9 F2 22 85<E9>
0980:EA A5 E9 F9 EA 22 85 E9<67>
0988:A5 E8 F9 E2 22 85 E8 A5<40>
0990:E7 F9 DA 22 85 E7 A5 E6<98>
0998:F9 D2 22 85 E6 A5 E5 F9<B4>
```

```
09A0:CA 22 85 E5 26 DA C8 C0<13>
09A8:08 D0 9E 4C B4 2D 36 D5<BE>
09B0:C8 C0 08 D0 F9 F0 F4 F8<FC>
09B8:18 A5 E5 8D F9 22 65 E5<83>
09C0:8D F8 22 A5 E4 8D F1 22<FE>
09C8:65 E4 8D F0 22 A5 E3 8D<AD>
09D0:E9 22 65 E3 8D E8 22 A5<8F>
09D8:E2 8D E1 22 65 E2 8D E0<23>
09E0:22 A5 E1 8D D9 22 65 E1<CC>
09E8:8D D8 22 A9 00 8D D1 22<3C>
09F0:69 00 8D D0 22 A2 02 BD<9D>
09F8:F6 22 7D F6 22 9D F5 22<2C>
0A00:BD EE 22 7D EE 22 9D ED<AA>
0A08:22 BD E6 22 7D E6 22 9D<A0>
0A10:E5 22 BD DE 22 7D DE 22<79>
0A18:9D DD 22 BD D6 22 7D D6<48>
0A20:22 9D D5 22 BD CE 22 7D<4F>
0A28:CE 22 9D CD 22 CA D0 C7<CB>
0A30:AD F8 22 6D F6 22 8D F5<E9>
0A38:22 AD F0 22 6D EE 22 8D<9F>
0A40:ED 22 AD E8 22 6D E6 22<15>
0A48:8D E5 22 AD E0 22 6D DE<1B>
0A50:22 8D DD 22 AD D8 22 6D<CC>
0A58:D6 22 8D D5 22 AD D0 22<54>
0A60:6D CE 22 8D CD 22 A2 02<51>
0A68:BD F3 22 7D F3 22 9D F2<EE>
0A70:22 BD EB 22 7D EB 22 9D<B7>
0A78:EA 22 BD E3 22 7D E3 22<C6>
0A80:9D E2 22 BD DB 22 7D DB<0F>
0A88:22 9D DA 22 BD D3 22 7D<28>
0A90:D3 22 9D D2 22 BD CB 22<FE>
0A98:7D CB 22 9D CA 22 CA 10<96>
0AA0:C7 A5 D5 85 E1 A5 D6 85<0E>
0AA8:E2 A5 D7 85 E3 A5 D8 85<EC>
0AB0:E4 A5 D9 85 E5 A9 00 85<92>
0AB8:D4 85 D5 85 D6 85 D7 85<C2>
0AC0:D8 85 D9 60 A9 05 85 F4<AB>
0AC8:A9 80 85 F3 60 EE 01 D3<60>
0AD0:20 A1 DB CE 01 D3 60 EE<57>
0AD8:01 D3 20 E6 D8 CE 01 D3<4F>
0AE0:60 EE 01 D3 20 00 D8 CE<FC>
0AE8:01 D3 60 A4 F2 B1 F3 38<EC>
0AF0:E9 30 C9 0A 60 A4 D4 A5<76>
0AF8:D5 85 F7 20 9C 2B F8 98<12>
0B00:F0 2E 4A 4A 4A 4A 85 F8<DD>
0B08:98 29 07 A0 00 90 02 69<4A>
0B10:07 46 F8 90 02 69 15 46<55>
0B18:F8 90 02 69 31 46 F8 90<F7>
0B20:05 69 63 90 01 C8 46 F8<E7>
0B28:90 06 69 27 C8 90 01 C8<71>
0B30:A6 F7 F0 6F 46 F7 90 07<FF>
0B38:69 55 C8 C8 90 01 C8 46<B3>
0B40:F7 90 08 69 11 AA 98 69<56>
0B48:05 A8 8A 46 F7 90 08 69<E3>
0B50:23 AA 98 69 10 A8 8A 46<88>
0B58:F7 90 08 69 47 AA 98 69<D0>
0B60:20 A8 8A 46 F7 90 08 69<38>
0B68:95 AA 98 69 40 A8 8A 46<F6>
0B70:F7 90 0C 69 91 AA 98 69<01>
0B78:81 A8 8A 90 02 E6 D5 46<3A>
0B80:F7 90 0E 69 83 AA 98 69<E1>
0B88:63 A8 A5 D5 69 01 85 D5<89>
0B90:8A 46 F7 90 0E 69 67 AA<E2>
0B98:98 69 27 A8 A5 D5 69 03<C2>
0BA0:85 D5 8A 84 D6 85 D7 A9<57>
0BA8:42 85 D4 4C B8 2D 18 60<5A>
0BB0:A5 E0 49 80 85 E0 A5 E0<07>
0BB8:29 7F F0 F2 85 F7 A5 D4<47>
0BC0:29 7F 38 E5 F7 B0 33 A5<E6>
0BC8:D4 A4 E0 85 E0 84 D4 A5<93>
0BD0:D5 A4 E1 85 E1 84 D5 A5<54>
0BD8:D6 A4 E2 85 E2 84 D6 A5<D5>
0BE0:D7 A4 E3 85 E3 84 D7 A5<77>
0BE8:D8 A4 E4 85 E4 84 D8 A5<32>
0BF0:D9 A4 E5 85 E5 84 D9 4C<FB>
0BF8:9D 2C A8 F0 4B 88 F0 36<8A>
0C00:88 F0 22 88 F0 10 88 D0<B5>
0C08:A5 A5 E1 85 E5 84 E4 84<09>
0C10:E3 84 E2 4C 2D 2D A5 E2<05>
0C18:85 E5 A5 E1 85 E4 84 E3<7E>
0C20:84 E2 4C 2D 2D A5 E3 85<92>
0C28:E5 A5 E2 85 E4 A5 E1 85<62>
0C30:E3 84 E2 4C 2D 2D A5 E4<E7>
0C38:85 E5 A5 E3 85 E4 A5 E2<7F>
0C40:85 E3 A5 E1 85 E2 84 E1<89>
0C48:F8 A5 D4 45 E0 30 3A 18<9C>
0C50:A5 D9 65 E5 85 D9 A5 D8<98>
0C58:65 E4 85 D8 A5 D7 65 E3<DF>
0C60:85 D7 A5 D6 65 E2 85 D6<F4>
0C68:A5 D5 65 E1 85 D5 90 16<F9>
0C70:A5 D8 85 D9 A5 D7 85 D8<E3>
0C78:A5 D6 85 D7 A5 D5 85 D6<2E>
0C80:A9 01 85 D5 E6 D4 4C B8<7A>
0C88:2D 38 A5 D9 E5 E5 85 D9<E6>
```

```
0C90:A5 D8 E5 E4 85 D8 A5 D7<F4>
0C98:E5 E3 85 D7 A5 D6 E5 E2<0A>
0CA0:85 D6 A5 D5 E5 E1 85 D5<5B>
0CA8:B0 27 A5 D4 49 80 85 D4<7B>
0CB0:38 98 E5 D9 85 D9 98 E5<92>
0CB8:D8 85 D8 98 E5 D7 85 D7<B6>
0CC0:98 E5 D6 85 D6 98 E5 D5<94>
0CC8:85 D5 4C B8 2D A6 DA D0<D1>
0CD0:02 A2 00 D8 A4 D4 F0 6D<99>
0CD8:A5 D5 D0 5C 88 A5 D6 D0<CE>
0CE0:47 88 A5 D7 D0 30 88 A5<3B>
0CE8:D8 D0 1B 88 A5 D9 D0 08<7C>
0CF0:88 8A F0 51 85 D5 D0 40<51>
0CF8:85 D5 86 D6 A9 00 85 D9<5D>
0D00:85 D8 85 D7 F0 32 85 D5<F7>
0D08:A5 D9 85 D6 86 D7 A9 00<1A>
0D10:85 D9 85 D8 F0 22 85 D5<37>
0D18:A5 D8 85 D6 A5 D9 85 D7<19>
0D20:86 D8 A9 00 85 D9 F0 10<09>
0D28:85 D5 A5 D7 85 D6 A5 D8<9C>
0D30:85 D7 A5 D9 85 D8 86 D9<EE>
0D38:84 D4 98 29 7F C9 71 B0<56>
0D40:08 C9 0F B0 03 20 9C 2B<7A>
0D48:18 60 86 FC 84 FD 85 EF<E0>
0D50:20 B5 2E 20 7C 2E C6 EF<7C>
0D58:20 9E 26 B0 37 A5 FC 69<16>
0D60:06 85 FC 90 02 E6 FD 20<D8>
0D68:7C 2E 20 9D 2C B0 25 C6<5A>
0D70:EF F0 21 AD E0 05 85 E0<1B>
0D78:AD E1 05 85 E1 AD E2 05<1D>
0D80:85 E2 AD E3 05 85 E3 AD<4A>
0D88:E4 05 85 E4 AD E5 05 85<2C>
0D90:E5 4C 3F 2E 60 A0 05 B1<37>
0D98:FC 85 E5 88 B1 FC 85 E4<4B>
0DA0:88 B1 FC 85 E3 88 B1 FC<87>
0DA8:85 E2 88 B1 FC 85 E1 88<F9>
0DB0:B1 FC 85 E0 60 A5 D4 85<93>
0DB8:E0 A5 D5 85 E1 A5 D6 85<E9>
0DC0:E2 A5 D7 85 E3 A5 D8 85<50>
0DC8:E4 A5 D9 85 E5 60 A5 D4<69>
0DD0:8D E0 05 A5 D5 8D E1 05<B3>
0DD8:A5 D6 8D E2 05 A5 D7 8D<A0>
0DE0:E3 05 A5 D8 8D E4 05 A5<76>
0DE8:D9 8D E5 05 60 A5 D4 8D<C1>
0DF0:E6 05 A5 D5 8D E7 05 A5<57>
0DF8:D6 8D E8 05 A5 D7 8D E9<92>
0E00:05 A5 D8 8D EA 05 A5 D9<21>
0E08:8D EB 05 60 A2 00 4C FF<F0>
0E10:2E A2 06 4C FF 2E A2 0C<63>
0E18:BD E0 05 85 D4 BD E1 05<CB>
0E20:85 D5 BD E2 05 85 D6 BD<BE>
0E28:E3 05 85 D7 BD E4 05 85<F9>
0E30:D8 BD E5 05 85 D9 60 A2<40>
0E38:0C 4C 25 2F A2 06 BD E0<85>
0E40:05 85 E0 BD E1 05 85 E1<ED>
0E48:BD E2 05 85 E2 BD E3 05<31>
0E50:85 E3 BD E4 05 85 E4 BD<20>
0E58:E5 05 85 E5 60 A2 05 EE<A0>
0E60:01 D3 BD B9 DE 95 E0 CA<A0>
0E68:10 F8 CE 01 D3 20 9E 26<CC>
0E70:B0 EA A9 00 85 F1 A5 D4<7B>
0E78:85 F0 29 7F 85 D4 C9 40<B1>
0E80:90 1E D0 D8 A5 D5 29 F0<9E>
0E88:4A 85 F1 4A 4A 65 F1 85<34>
0E90:F1 A5 D5 29 0F 65 F1 85<EC>
0E98:F1 A9 00 85 D5 20 B8 2D<73>
0EA0:A9 0A A2 4D A0 DE EE 01<ED>
0EA8:D3 20 31 2E CE 01 D3 20<77>
0EB0:9B 26 A5 F1 F0 0F 4A 18<CF>
0EB8:65 D4 30 16 85 D4 46 F1<7C>
0EC0:90 03 20 9B 32 06 F0 90<2C>
0EC8:0A 20 9C 2E 20 BB 2F 4C<DE>
0ED0:00 28 38 60 A9 40 85 D4<19>
0ED8:A0 01 84 D5 88 84 D6 84<64>
0EE0:D7 84 D8 84 D9 60 86 FC<D9>
0EE8:84 FD 20 B5 2E 20 7C 2E<ED>
0EF0:20 9D 2C 20 D4 2E 20 F3<55>
0EF8:2E 20 7C 2E 20 97 2C 20<59>
0F00:23 2F 4C 00 28 38 60 A9<70>
0F08:05 D0 02 A9 00 85 F0 A5<FE>
0F10:D4 30 F2 F0 F0 0A 49 80<1B>
0F18:85 F1 A9 40 85 D4 A5 D5<E0>
0F20:29 F0 F0 05 E6 F1 20 A3<F3>
0F28:32 A2 66 A0 DF EE 01 D3<22>
0F30:20 CD 2F 20 D4 2E 20 9B<AC>
0F38:26 A9 0A A2 72 A0 DF 20<E6>
0F40:31 2E CE 01 D3 20 23 2F<B8>
0F48:20 9E 26 A9 3F 85 E0 A9<91>
0F50:50 85 E1 A9 00 85 E2 85<89>
0F58:E3 85 E4 85 E5 20 9D 2C<EA>
0F60:20 9C 2E A5 F1 10 05 18<2F>
0F68:49 FF 69 01 85 D4 A9 00<F8>
0F70:85 D5 20 DC 2B A5 F1 29<3F>
0F78:80 05 D4 85 D4 20 9D 2C<71>
0F80:A6 F0 F0 11 EE 01 D3 BD<D3>
0F88:89 DE 95 E0 CA 10 F8 CE<E0>
0F90:01 D3 4C 00 28 18 60 38<07>
0F98:60 A9 04 24 D4 10 06 A9<67>
0FA0:02 D0 02 A9 01 85 F0 A5<2D>
0FA8:D4 29 7F 85 D4 A6 FB BD<FB>
0FB0:36 31 85 E0 BD 37 31 85<6E>
0FB8:E1 BD 38 31 85 E2 BD 39<28>
0FC0:31 85 E3 BD 3A 31 85 E4<44>
0FC8:BD 3B 31 85 E5 20 00 28<52>
0FD0:B0 C5 A5 D4 29 7F 38 E9<A6>
0FD8:40 30 24 C9 04 10 B8 AA<2B>
0FE0:B5 D5 85 F1 29 10 F0 02<B9>
0FE8:A9 02 18 65 F1 29 03 65<3B>
0FF0:F0 85 F0 86 F1 A9 00 95<90>
0FF8:D5 CA 10 FB 20 B8 2D 46<65>
1000:F0 90 09 20 9C 2E 20 BB<F0>
1008:2F 20 97 2C 20 D4 2E 20<09>
1010:9B 26 B0 83 A9 06 A2 18<3E>
1018:A0 31 20 31 2E 20 23 2F<D2>
1020:20 9E 26 46 F0 90 09 18<FC>
1028:A5 D4 F0 04 49 80 85 D4<F1>
1030:60 BD 03 55 14 99 39 3E<C8>
1038:01 60 44 27 52 BE 46 81<81>
1040:75 43 55 3F 07 96 92 62<1F>
1048:39 BF 64 59 64 08 67 40<BE>
1050:01 57 07 96 32 40 90 00<72>
1058:00 00 00 3F 01 74 53 29<15>
1060:25 A9 00 85 F0 85 F1 A5<E2>
1068:D4 29 7F C9 40 30 1B A5<33>
1070:D4 29 80 85 F0 E6 F1 A9<0F>
1078:7F 25 D4 85 D4 A2 EA A0<1D>
1080:DF EE 01 D3 20 CD 2F CE<39>
1088:01 D3 20 D4 2E 20 9B 26<54>
1090:B0 47 A9 0B A2 AE A0 DF<FE>
1098:EE 01 D3 20 31 2E CE 01<2B>
10A0:D3 B0 36 20 23 2F 20 9E<8E>
10A8:26 B0 2E A5 F1 F0 19 A2<F3>
10B0:05 EE 01 D3 BD F0 DF 95<14>
10B8:E0 CA 10 F8 CE 01 D3 20<35>
10C0:9D 2C A5 F0 05 D4 85 D4<22>
10C8:A5 FB F0 0D A2 05 BD 42<AD>
10D0:31 95 E0 CA 10 F8 20 00<A7>
10D8:28 60 38 60 18 60 A9 00<35>
10E0:85 F1 A5 D4 30 F4 F0 F4<36>
10E8:C9 3F F0 05 18 69 01 85<F4>
10F0:F1 A9 06 85 EF A9 3F 85<29>
10F8:D4 20 D4 2E 20 9C 2E 20<4F>
1100:BB 2F E6 D5 20 97 2C 20<69>
1108:23 2F 20 9E 26 A5 D4 8D<1D>
1110:EC 05 A5 D5 8D ED 05 A5<3A>
1118:D6 8D EE 05 A5 D7 8D EF<E0>
1120:05 A5 D8 8D F0 05 A5 D9<A7>
1128:8D F1 05 20 9C 2E 20 FB<3D>
1130:2E 20 00 28 20 1E 2F 20<EB>
1138:97 2C 20 4F 32 A5 D4 F0<65>
1140:0C 20 1E 2F 20 9D 2C C6<B4>
1148:EF 10 C2 30 03 20 FD 2E<F1>
1150:A5 F1 F0 0F 4A 18 65 D4<51>
1158:E9 1F 85 D4 46 F1 90 03<9E>
1160:20 9B 32 18 60 4C 9C 2B<C7>
1168:A5 D4 29 7F C9 0F 90 F5<06>
1170:20 9C 2E F8 A2 00 A0 04<1F>
1178:18 A5 D9 65 E5 85 D9 A5<C5>
1180:D8 65 E4 85 D8 A5 D7 65<22>
1188:E3 85 D7 A5 D6 65 E2 85<2F>
1190:D6 A5 D5 65 E1 85 D5 8A<04>
1198:69 00 AA 88 D0 DB D8 8A<50>
11A0:F0 1A A5 D8 85 D9 A5 D7<E3>
11A8:85 D8 A5 D6 85 D7 A5 D5<B4>
11B0:85 D6 86 D5 A5 D5 C9 10<CA>
11B8:90 31 E6 D4 A5 D5 C9 10<0A>
11C0:90 27 4A 66 D6 66 D7 66<6C>
11C8:D8 66 D9 4A 66 D6 66 D7<4E>
11D0:66 D8 66 D9 4A 66 D6 66<CB>
11D8:D7 66 D8 66 D9 4A 66 D6<E2>
11E0:66 D7 66 D8 66 D9 85 D5<A5>
11E8:60 C6 D4 A9 00 06 D9 26<A0>
11F0:D8 26 D7 26 D6 26 D5 2A<1D>
11F8:06 D9 26 D8 26 D7 26 D6<44>
1200:26 D5 2A 06 D9 26 D8 26<47>
1208:D7 26 D6 26 D5 2A 06 D9<92>
1210:26 D8 26 D7 26 D6 26 D5<41>
1218:2A 60 C9 FF 90 02 C0 50<CE>
1220:8A 69 00 85 D4 60 98 29<34>
1228:F0 4A 85 D4 4A 4A 65 D4<BA>
1230:85 D4 98 29 0F A6 D6 E0<BD>
1238:50 65 D4 85 D4 60 A2 00<89>
1240:A4 D5 A5 D4 86 D5 38 E9<26>
1248:40 90 CF F0 D9 C9 02 F0<EE>
1250:45 B0 42 A5 D6 29 F0 4A<8E>
1258:85 D4 4A 4A 65 D4 85 D4<EB>
1260:A5 D6 29 0F A6 D7 E0 50<7F>
1268:65 D4 85 D4 98 29 0F AA<33>
1270:BD CD 33 65 D4 85 D4 BD<6A>
1278:D7 33 65 D5 85 D5 98 29<90>
1280:F0 F0 12 4A 4A 4A AA A5<EB>
1288:D4 7D B9 33 85 D4 A5 D5<CD>
1290:7D BA 33 85 D5 60 C0 07<43>
1298:B0 FB A5 D7 29 F0 4A 85<77>
12A0:D4 4A 4A 65 D4 85 D4 A5<95>
12A8:D7 29 0F A6 D8 E0 50 65<D2>
12B0:D4 85 D4 B9 AD 33 65 D4<B2>
12B8:85 D4 B9 B4 33 65 D5 85<C0>
12C0:D5 A4 D6 D0 A7 60 00 10<FE>
12C8:20 30 40 50 60 00 27 4E<6D>
12D0:75 9C C3 EA E8 03 D0 07<00>
12D8:B8 0B A0 0F 88 13 70 17<F8>
12E0:58 1B 40 1F 28 23 00 64<12>
12E8:C8 2C 90 F4 58 BC 20 84<22>
12F0:00 00 00 01 01 01 02 02<36>
12F8:03 03 4C 28 E5 20 1E F4<20>
1300:20 05 E5 85 A2 84 A3 05<D5>
1308:A3 F0 EF 98 05 9A 05 9C<87>
1310:30 E8 A5 99 85 A0 A5 9A<0C>
1318:85 A1 20 62 C9 A5 8A 85<77>
1320:E2 A5 8B 85 E3 A9 80 85<6B>
1328:D5 0A 85 D4 20 0E 35 30<DC>
1330:C9 A5 89 85 8B A5 88 85<EA>
1338:8A A0 01 B1 8A 30 44 C8<93>
1340:B1 8A 85 9F C8 B1 8A 85<66>
1348:A7 C8 84 A8 B1 8A C9 0A<3A>
1350:F0 0A C9 0B F0 06 C9 0C<F1>
1358:F0 02 C9 0D F0 69 C9 1E<B7>
1360:F0 6B C9 23 F0 5A C9 07<66>
1368:F0 4D C9 04 F0 6B C9 56<45>
1370:F0 67 A4 A7 C4 9F 90 CD<EE>
1378:18 A5 8A 65 9F 90 B8 E6<E8>
1380:8B B0 B4 A5 E3 85 8B A5<02>
1388:E2 85 8A A0 01 B1 8A 30<2F>
1390:20 A5 9C 91 8A 88 A5 9B<0F>
1398:91 8A 18 65 A2 85 9B A5<F2>
13A0:9C 65 A3 85 9C A0 02 B1<B1>
13A8:8A 65 8A 90 DC E6 8B B0<72>
13B0:D8 20 C5 C9 4C B5 F1 20<2F>
13B8:66 35 E0 1B D0 B4 C6 A8<0D>
13C0:A4 A8 C8 C4 A7 B0 AB 20<65>
13C8:D5 34 4C 59 34 20 66 35<97>
13D0:E0 17 F0 07 E0 18 D0 9A<B4>
13D8:2C E6 A8 A5 A8 C5 A7 B0<AA>
13E0:91 48 20 D7 34 68 85 A8<42>
13E8:20 68 35 4C C2 34 E6 A8<E6>
13F0:A4 A8 84 DC B1 8A F0 2E<42>
13F8:C9 0F B0 2A 20 4F E4 20<21>
1400:25 33 A5 D5 30 20 B0 1E<C8>
1408:20 0E 35 08 B0 04 85 D4<57>
1410:84 D5 20 DC 2B 06 D4 28<B2>
1418:66 D4 A4 AB A2 05 B5 D4<D6>
1420:88 91 8A CA 10 F8 60 A5<14>
1428:D4 C5 99 A5 D5 E5 9A B0<0E>
1430:05 A5 D4 A4 D5 60 A5 E2<30>
1438:85 DA A5 E3 85 DB A5 9B<53>
1440:85 E0 A5 9C 85 E1 A0 01<01>
1448:B1 DA 30 2D C5 D5 D0 05<5E>
1450:88 B1 DA C5 D4 B0 1F A0<3B>
1458:02 B1 DA 65 DA 85 DA 90<A4>
1460:02 E6 DB 88 B1 DA 30 11<57>
1468:18 A5 E0 65 A2 85 E0 A5<A8>
1470:E1 65 A3 4C 2B 35 18 F0<E9>
1478:01 38 A5 E0 A4 E1 60 E6<11>
1480:A8 20 4F E4 90 FB AA BD<65>
1488:CC 23 D0 F5 60 A9 00 2C<7A>
1490:A9 80 8D 16 23 A9 F6 A0<7B>
1498:35 8D E0 02 8C E1 02 A9<6F>
14A0:04 A0 01 20 C8 C2 EE 01<37>
14A8:D3 20 07 36 C9 FF D0 60<88>
14B0:C8 D0 5D A9 F6 A0 35 8D<0C>
14B8:E2 02 8C E3 02 20 07 36<C6>
14C0:C9 FF D0 04 C0 FF F0 EB<0C>
14C8:9D 44 03 98 9D 45 03 20<20>
14D0:07 36 38 FD 44 03 9D 48<50>
14D8:03 98 FD 45 03 9D 49 03<BB>
14E0:FE 48 03 D0 03 FE 49 03<1C>
14E8:20 56 E4 98 30 24 20 01<0D>
14F0:36 AD 53 03 C9 03 D0 BB<8D>
14F8:A2 10 A9 0C 9D 42 03 20<5F>
1500:56 E4 2C 16 23 10 03 20<B4>
1508:04 36 A9 FE 8D 01 D3 60<04>
1510:A9 01 A8 20 24 23 98 4C<EE>
1518:E4 DC 6C E2 02 6C E0 02<56>
1520:A9 07 A2 10 9D 42 03 A9<52>
1528:00 9D 48 03 9D 49 03 20<06>
1530:56 E4 30 DF 48 A9 00 9D<97>
1538:48 03 20 56 E4 30 D4 A8<55>
1540:68 60 00 60 78 61 AE 24<14>
```

Listing »Turbo-Basic XL« (Fortsetzung)

```
1548:E4 AC 25 E4 E8 D0 01 C8<7A>
1550:8E F8 24 8C F9 24 A9 00<56>
1558:8D 0E D4 78 A9 FE 8D 01<28>
1560:D3 A9 AB 8D FA FF A9 24<0F>
1568:8D FB FF A9 D0 8D FE FF<32>
1570:A9 24 8D FF FF A9 CC 20<4C>
1578:DB 60 A9 E0 20 DB 60 A9<33>
1580:40 8D 0E D4 58 A9 FF 8D<1D>
1588:01 D3 20 01 61 A9 00 85<2E>
1590:09 A5 0C A4 0D 8D 8E 20<FF>
1598:8C 8F 20 A9 80 A0 20 85<52>
15A0:0C 84 0D A9 FE 8D 01 D3<25>
15A8:85 CA A2 01 8E F8 03 86<C4>
15B0:09 CA 8E 44 02 20 04 E6<42>
15B8:A9 00 85 A0 85 A1 20 5F<0F>
15C0:C9 A0 1F A2 8A 20 86 25<74>
15C8:A0 1E B9 A7 60 91 97 88<D6>
15D0:10 F8 20 C5 C9 A9 FF 48<07>
15D8:A9 AE 48 20 C6 60 CE 01<95>
15E0:D3 A9 00 A8 99 00 B0 88<4C>
15E8:10 FA 4C 3A F5 00 00 19<66>
15F0:07 0D 67 14 19 25 0F 0D<93>
15F8:44 3A 41 55 54 4F 52 55<15>
1600:4E 2E 42 41 53 16 01 00<10>
1608:06 06 16 16 A9 FF 8D 01<3A>
1610:D3 A9 C0 85 6A 4A 85 0F<A2>
1618:AD 01 E4 48 AD 00 E4 48<51>
1620:60 85 D5 A0 00 84 D4 A2<8B>
1628:04 A9 FF 8D 01 D3 B1 D4<C3>
1630:99 00 21 C8 D0 F8 CE 01<4C>
1638:D3 B9 00 21 91 D4 C8 D0<83>
1640:F8 E6 D5 CA D0 E3 60 A2<AC>
1648:10 A9 D6 9D 44 03 A9 00<BA>
1650:9D 45 03 9D 49 03 A9 04<EB>
1658:9D 48 03 A9 07 9D 42 03<86>
1660:20 56 E4 30 59 A9 7C 85<8F>
1668:DA 9D 44 03 A9 61 85 DB<56>
1670:9D 45 03 A5 D8 E5 D6 85<EA>
1678:DC A5 D9 E5 D7 85 DD E6<D3>
1680:DC D0 02 E6 DD A5 DC 9D<FF>
1688:48 03 A5 DD 9D 49 03 20<2C>
1690:56 E4 30 2B CE 01 D3 A0<53>
1698:00 A6 DD F0 0E B1 DA 91<82>
16A0:D6 C8 D0 F9 E6 D7 E6 DB<64>
16A8:CA D0 F2 A6 DC F0 08 B1<A5>
16B0:DA 91 D6 C8 CA D0 F8 A9<8B>
16B8:FF 8D 01 D3 30 89 60 6C<A5>
16C0:0A 00 E2 02 E3 02 00 60<CD>
16C8:00 C0 F7 CB 20 1E F4 20<F1>
16D0:CB C0 20 FB E4 85 A0 84<FE>
16D8:A1 20 FB E4 20 5F C9 A0<26>
16E0:01 B1 8A C5 D5 D0 05 88<EE>
16E8:B1 8A C5 D4 90 02 D0 16<5F>
16F0:A0 02 18 B1 8A A8 65 8A<40>
16F8:85 8A 90 02 E6 8B A2 8A<A2>
1700:20 20 26 4C 13 C0 20 26<89>
1708:C1 4C B5 F1 C8 C4 A7 B0<AB>
1710:03 20 BE C2 A9 00 85 B9<84>
1718:20 94 C2 A5 11 F0 34 A5<63>
1720:B9 20 B3 E4 A5 9D C5 88<F4>
1728:A5 9E E5 89 B0 27 A5 B9<51>
1730:20 6B C0 E6 B9 D0 E1 20<56>
1738:B1 F3 20 02 F2 A5 D2 C9<2E>
1740:C0 B0 1E C9 80 B0 38 C9<C6>
1748:40 B0 34 A9 3D 20 96 C2<6A>
1750:4C 59 F9 C6 11 A5 B5 F0<6A>
1758:07 20 F8 C4 A9 00 85 B5<62>
1760:60 A0 4F 4A B0 0A A0 5D<77>
1768:4A B0 05 A9 3F 4C 96 C2<EB>
1770:98 20 4B F3 A0 00 B1 D4<BF>
1778:AA C8 B1 D4 4C 52 F9 A5<7F>
1780:D9 48 A5 D8 48 A5 D7 A6<71>
1788:D6 20 52 F9 A9 2C 20 96<4F>
1790:C2 68 AA 68 4C 52 F9 8A<88>
1798:48 A5 90 48 A5 91 48 A5<2D>
17A0:B2 48 A5 A0 48 A5 A1 48<A6>
17A8:20 D3 F7 B0 2F A4 A1 D0<8E>
17B0:08 A9 80 85 A0 A9 C1 85<0B>
17B8:A1 A8 30 0E A0 00 B1 A0<0E>
17C0:C8 91 90 B1 A0 C8 91 90<03>
17C8:90 DE A0 00 B1 A0 A0 13<8A>
17D0:91 90 A0 01 B1 A0 A0 14<EA>
17D8:91 90 90 CC 68 85 A1 68<A4>
17E0:85 A0 68 85 B2 68 85 91<03>
17E8:85 0F 68 85 90 85 0E 68<EC>
17F0:AA 60 48 A5 90 48 A5 91<24>
17F8:48 A5 B2 48 A5 A0 48 A5<47>
1800:A1 48 A5 8A 48 A5 8B 48<FB>
1808:A5 BE 48 A5 BF 48 20 82<52>
1810:C1 20 C5 C9 20 D3 F7 B0<C8>
1818:24 48 A4 A1 C8 F0 05 20<23>
1820:62 C9 90 06 A9 00 85 8A<9E>
1828:85 8B A0 01 68 10 02 A0<A9>
1830:13 A5 8A 91 90 C8 A5 8B<CD>
1838:91 90 4C 48 C1 68 85 BF<04>
1840:68 85 BE 68 85 8B 68 85<3D>
1848:8A 4C 10 C1 FF FF A5 86<F3>
1850:85 A0 A5 87 85 A1 A5 A0<14>
1858:C5 88 A5 A1 E5 89 B0 19<BD>
1860:A0 00 B1 A0 29 C0 C9 C0<AA>
1868:D0 02 91 A0 18 A5 A0 69<3A>
1870:08 85 A0 90 E1 E6 A1 B0<62>
1878:DD 60 20 05 E5 29 0F 06<73>
1880:9B 06 9B 06 9B 06 9B 05<DC>
1888:9B A6 9A D0 15 A6 99 E0<3C>
1890:05 B0 0F 9D C4 02 60 A2<B2>
1898:07 A9 00 9D 00 D2 CA 10<6F>
18A0:FA 60 4C 28 E5 85 A2 C8<2A>
18A8:C4 A7 B0 EB 20 23 E5 A0<20>
18B0:00 24 A2 10 03 0A A0 78<36>
18B8:C9 04 B0 E6 0A 48 8C 08<31>
18C0:D2 A9 03 8D 0F D2 20 05<42>
18C8:E5 68 AA A5 99 9D 00 D2<EE>
18D0:24 A2 10 07 E8 E8 A5 9A<F1>
18D8:9D 00 D2 A5 9B 0A 0A 0A<86>
18E0:0A 05 D4 9D 01 D2 60 20<D8>
18E8:13 E5 85 55 84 56 20 23<91>
18F0:E5 85 54 60 20 13 E5 85<10>
18F8:C8 60 20 13 E5 8D FD 02<B2>
1900:60 A9 12 2C A9 11 48 20<C3>
1908:1B C2 A5 C8 8D FB 02 A2<87>
1910:60 A9 0C 9D 4A 03 A9 00<91>
1918:9D 4B 03 68 4C FE C5 A2<C8>
1920:06 86 C1 20 F8 C4 20 13<25>
1928:E5 A2 0D A0 23 86 F3 84<AA>
1930:F4 A2 06 29 F0 49 1C A8<A0>
1938:A5 D4 20 B4 C3 4C B9 C4<73>
1940:20 1B C2 A4 C8 A2 60 4C<1C>
1948:9C C2 A6 B4 D0 05 A5 C2<8E>
1950:20 96 C2 A6 B4 A9 05 20<49>
1958:AD C2 20 02 C5 4C B9 C4<82>
1960:A9 9B A6 B5 A8 20 AF C2<FB>
1968:BD 4A 03 85 2A BD 4B 03<52>
1970:85 2B 20 DE 24 98 4C BF<E7>
1978:C4 85 C0 86 C1 4C AB C4<9E>
1980:A9 04 20 C6 C2 85 B4 4C<45>
1988:7E E6 A9 08 20 C6 C2 85<29>
1990:B5 60 A0 07 84 C1 48 20<71>
1998:AB C4 20 FD C4 A0 03 84<0B>
19A0:C0 68 A0 00 20 DC C3 A9<1F>
19A8:07 60 A9 FF 2C A9 00 48<CB>
19B0:A9 04 20 C6 C2 68 48 A9<85>
19B8:07 85 C0 85 CA 20 AB C4<C8>
19C0:A0 0E 20 04 C5 20 B9 C4<66>
19C8:AD 80 05 0D 81 05 D0 38<DA>
19D0:A2 8C 18 A5 80 7D 00 05<BE>
19D8:A8 A5 81 7D 01 05 CD E6<2B>
19E0:02 90 0A D0 05 CC E5 02<98>
19E8:90 03 4C D4 F8 95 01 94<94>
19F0:00 CA CA E0 82 B0 DB 20<74>
19F8:6E C3 20 5B F5 A9 00 85<9B>
1A00:CA 68 F0 01 60 4C 6B E6<30>
1A08:A9 00 85 CA 4C D1 F8 A9<08>
1A10:08 20 C6 C2 A9 0B 85 C0<61>
1A18:A2 80 38 B5 00 E5 80 9D<AF>
1A20:00 05 E8 B5 00 E5 81 9D<1F>
1A28:00 05 E8 E0 8E 90 EB 20<45>
1A30:AB C4 A0 0E 20 04 C5 20<52>
1A38:B9 C4 20 AB C4 A5 82 85<E9>
1A40:F3 A5 83 85 F4 AC 8D 05<2E>
1A48:88 98 AC 8C 05 20 06 C5<2E>
1A50:20 B9 C4 4C F8 C4 A9 08<40>
1A58:20 9C C3 4C 48 C3 A9 04<53>
1A60:20 9C C3 A9 00 4C EA C2<45>
1A68:48 A2 10 86 F3 A2 23 86<AC>
1A70:F4 A2 07 68 A8 A9 80 20<6F>
1A78:B4 C3 20 B9 C4 A9 07 60<39>
1A80:48 A9 03 20 AD C2 68 9D<3D>
1A88:4B 03 98 9D 4A 03 20 0D<AB>
1A90:C5 4C AB 2B 20 13 E5 2C<75>
1A98:A9 03 85 C0 20 A6 C4 20<DF>
1AA0:13 E5 48 20 13 E5 A8 68<67>
1AA8:48 98 48 A4 A8 C8 C4 A7<7E>
1AB0:B0 1A 20 00 E4 20 73 C5<52>
1AB8:20 AB C4 68 9D 4B 03 68<22>
1AC0:9D 4A 03 20 02 C5 20 93<32>
1AC8:C5 4C B9 C4 A9 07 A2 23<8D>
1AD0:A0 05 20 71 DA 4C E9 C3<39>
1AD8:20 A6 C4 A9 0D 20 1A C5<8A>
1AE0:98 4C 20 C5 A9 26 20 51<32>
1AE8:C5 BD 4C 03 BC 4D 03 20<21>
1AF0:22 C5 20 AB C4 BD 4E 03<C5>
1AF8:4C 20 C5 20 A6 C4 20 13<84>
1B00:E5 20 AB C4 A5 D4 9D 4C<45>
1B08:03 A5 D5 9D 4D 03 20 13<25>
1B10:E5 20 AB C4 A5 D4 9D 4E<37>
1B18:03 A9 25 85 C0 4C 56 C5<C8>
1B20:20 9C C4 20 13 E5 A6 C1<FC>
1B28:20 98 C2 A4 A8 C8 C4 A7<55>
1B30:90 F1 60 B1 8A C9 1C F0<21>
1B38:0E 20 0D C6 20 20 C5 A4<2A>
1B40:A8 C8 C4 A7 90 F3 60 20<96>
1B48:A6 C4 20 AB C4 20 04 C6<65>
1B50:AA 98 20 BF C4 8A 20 20<5C>
1B58:C5 A4 A8 C8 C4 A7 90 EA<8F>
1B60:60 20 1B C2 A2 60 D0 E5<6D>
1B68:B1 8A C9 1C F0 04 A9 00<97>
1B70:F0 03 20 78 DD 85 C1 A5<EE>
1B78:C1 0A 0A 0A 0A AA 10 E0<6A>
1B80:A9 14 4C DC F8 20 AB C4<B0>
1B88:BD 43 03 10 D3 A0 00 8C<F0>
1B90:FE 02 C9 80 D0 09 84 11<34>
1B98:A6 CA F0 C4 4C 5D E6 A4<9F>
1BA0:C1 C9 88 F0 0F 85 B9 C0<37>
1BA8:07 D0 03 20 F8 C4 20 D1<9C>
1BB0:F5 4C DE F8 C0 07 D0 ED<66>
1BB8:A2 5D E4 C2 D0 E7 20 F8<D1>
1BC0:C4 4C 6E E6 20 AB C4 F0<3F>
1BC8:97 A9 0C 4C 1A C5 A0 FF<1D>
1BD0:A9 00 9D 49 03 98 9D 48<EA>
1BD8:03 A5 F4 A4 F3 9D 45 03<4B>
1BE0:98 9D 44 03 A5 C0 9D 42<EB>
1BE8:03 4C 98 24 A0 00 48 98<0D>
1BF0:48 20 00 E4 68 85 D5 68<F7>
1BF8:85 D4 20 DC 2B 4C C0 E5<90>
1C00:A9 00 A2 07 9D 00 D2 CA<8D>
1C08:10 FA A0 07 84 C1 20 F8<AA>
1C10:C4 C6 C1 D0 F9 60 C8 C4<4D>
1C18:A7 B0 E5 A9 0C 85 C0 20<C8>
1C20:A6 C4 20 18 C5 4C B9 C4<12>
1C28:A2 06 86 F2 BD 6C C5 20<C5>
1C30:96 C2 A6 F2 CA 10 F3 60<6C>
1C38:9B 59 44 41 45 52 9B 20<29>
1C40:DC E4 A5 D4 85 F3 A5 D5<47>
1C48:85 F4 A4 D6 A6 D7 F0 02<A4>
1C50:A0 FF B1 F3 85 97 84 98<FD>
1C58:A9 9B 91 F3 85 92 60 A4<3E>
1C60:98 A5 97 91 F3 A9 00 85<B9>
1C68:92 4C AB 2B A9 06 20 C6<68>
1C70:C2 A2 70 20 04 C6 30 07<C0>
1C78:A2 00 20 98 C2 10 F2 98<D9>
1C80:48 A2 70 20 FD C4 68 C9<E6>
1C88:88 F0 03 4C DC F8 60 A9<CE>
1C90:20 2C A9 23 2C A9 24 2C<C7>
1C98:A9 21 85 C0 A9 07 85 C1<4D>
1CA0:A9 00 A8 4C DC C3 A9 0B<2D>
1CA8:2C A9 07 48 20 A6 C4 20<C7>
1CB0:0C E5 20 AB C4 A5 9B 9D<5D>
1CB8:44 03 A5 9C 9D 45 03 A5<4E>
1CC0:D4 9D 48 03 A5 D5 9D 49<41>
1CC8:03 68 20 1A C5 4C A9 C2<A1>
1CD0:A9 07 85 C0 A0 00 4C 04<C4>
1CD8:C5 20 F4 24 C0 80 B0 01<1C>
1CE0:60 4C A9 C2 A9 0B 2C A9<0F>
1CE8:07 85 C0 20 9C C4 20 6B<76>
1CF0:E5 20 AB C4 A9 D4 85 F3<E6>
1CF8:A9 00 85 F4 A0 06 20 04<CD>
1D00:C5 20 AB 2B 20 B9 C4 A5<DD>
1D08:C0 C9 07 D0 03 20 C0 E5<2F>
1D10:A4 A8 C8 C4 A7 90 D7 60<E0>
1D18:A4 A9 D0 32 C6 AA A6 AA<37>
1D20:C6 AA BD 20 21 20 ED E5<99>
1D28:A0 02 BD 41 21 91 9D C8<AA>
1D30:BD 61 21 91 9D C8 BD 81<5E>
1D38:21 91 9D C8 BD A1 21 91<56>
1D40:9D C8 BD C1 21 91 9D C8<FB>
1D48:BD E1 21 91 9D 60 A9 80<FA>
1D50:85 B1 60 E6 B0 A4 A9 68<CB>
1D58:68 4C 31 E4 A9 40 85 B1<3C>
1D60:24 B1 10 06 A5 AA 85 AF<C3>
1D68:C6 AA A5 B0 A8 F0 07 C6<A0>
1D70:B0 20 16 E5 30 19 84 98<20>
1D78:85 97 20 16 E5 30 10 85<01>
1D80:F5 84 F6 20 6E E5 24 B1<CD>
1D88:50 08 A9 00 85 B1 60 4C<FD>
1D90:2B E5 46 D2 90 F9 A5 F5<0E>
1D98:C5 D6 A5 F6 E5 D7 B0 EF<8F>
1DA0:A5 97 C5 D8 A5 98 E5 D9<17>
1DA8:B0 E5 A5 D9 D0 10 A4 D9<F5>
1DB0:88 D0 0B A5 F5 85 F7 A5<AB>
1DB8:F6 85 F8 4C 14 C7 A0 00<BF>
1DC0:84 F7 84 F8 A0 10 46 D9<13>
1DC8:66 D8 90 0D 18 A5 F7 65<B4>
1DD0:F5 85 F7 A5 F8 65 F6 85<E1>
1DD8:F8 06 F5 26 F6 88 D0 E6<D9>
1DE0:18 A5 97 65 F7 85 F5 A5<D5>
1DE8:98 65 F8 06 F5 2A 85 F6<36>
1DF0:A8 A5 F5 0A 26 F6 65 F5<A9>
1DF8:AA 98 65 F6 A8 8A 65 D4<43>
1E00:AA 98 65 D5 A8 8A 65 8C<D3>
1E08:85 F5 98 65 8D 85 F6 24<B7>
1E10:B1 10 2E A6 AF 86 AA C6<77>
```

```
1E18:AA A0 00 84 B1 BD 40 21<79>
1E20:91 F5 C8 BD 60 21 91 F5<5E>
1E28:C8 BD 80 21 91 F5 C8 BD<68>
1E30:A0 21 91 F5 C8 BD C0 21<9B>
1E38:91 F5 C8 BD E0 21 91 F5<6A>
1E40:60 E6 AA A6 AA A0 05 B1<DC>
1E48:F5 9D E0 21 88 B1 F5 9D<3E>
1E50:C0 21 88 B1 F5 9D A0 21<AF>
1E58:88 B1 F5 9D 80 21 88 B1<BE>
1E60:F5 9D 60 21 88 B1 F5 9D<06>
1E68:40 21 5E 00 21 60 A5 B0<5C>
1E70:F0 07 20 0F C8 84 98 85<5F>
1E78:97 20 0F C8 38 E9 01 85<4E>
1E80:F5 98 E9 00 85 F6 20 6E<EC>
1E88:E5 A5 B1 10 0B 05 B0 85<E6>
1E90:B1 A4 D9 A5 D8 4C D0 C7<CA>
1E98:A5 D6 A4 D7 A6 B0 F0 10<6A>
1EA0:C6 B0 C4 98 90 3D D0 04<C2>
1EA8:C5 97 90 37 A4 98 A5 97<08>
1EB0:38 E5 F5 85 D6 AA 98 E5<2C>
1EB8:F6 85 D7 90 26 A8 D0 03<57>
1EC0:8A F0 20 20 DF E4 18 A5<3F>
1EC8:D4 65 F5 85 D4 A5 D5 65<C2>
1ED0:F6 85 D5 24 B1 10 01 60<6D>
1ED8:4C 3A E5 20 16 E5 D0 F7<02>
1EE0:AA D0 F4 A9 05 4C DC F8<87>
1EE8:20 DC E4 A5 D4 85 99 A5<7E>
1EF0:D5 85 9A A5 D6 85 A2 A4<23>
1EF8:D7 84 A3 A4 A9 F0 0F A9<0B>
1F00:80 85 B1 20 31 E4 A5 D7<E9>
1F08:A4 D6 26 B1 B0 07 20 DC<23>
1F10:E4 A5 D9 A4 D8 C5 A3 90<B6>
1F18:06 D0 08 C4 A2 B0 04 85<EB>
1F20:A3 84 A2 18 A5 D4 85 9B<60>
1F28:65 A2 A8 A5 D5 85 9C 65<BD>
1F30:A3 AA 38 98 E5 8C 85 F9<14>
1F38:8A E5 8D 85 FA 20 5D 26<8F>
1F40:A5 D3 20 B3 E4 38 A5 F9<F4>
1F48:E5 D4 A8 A5 FA E5 D5 AA<74>
1F50:A9 02 25 B1 F0 0F A9 00<A0>
1F58:85 B1 E4 D7 90 06 D0 05<B6>
1F60:C4 D6 B0 01 60 84 D6 86<7E>
1F68:D7 4C C0 E5 A4 A8 C4 A7<79>
1F70:90 01 60 20 00 E4 46 D2<13>
1F78:B0 67 38 26 D2 30 65 A5<80>
1F80:F5 69 01 85 F5 85 D6 A5<D6>
1F88:F6 69 00 30 54 85 D7 85<02>
1F90:F6 A5 97 69 01 85 D8 A5<0F>
1F98:98 69 00 85 D9 30 42 A0<F1>
1FA0:00 84 F7 84 F8 A0 10 A5<1A>
1FA8:F5 4A 90 0F 18 A5 F7 65<7D>
1FB0:D8 85 F7 A5 F8 65 D9 85<F1>
1FB8:F8 30 26 66 F8 66 F7 66<B7>
1FC0:F6 66 F5 88 D0 E1 06 F5<7D>
1FC8:26 F6 30 15 A6 F6 A5 F5<31>
1FD0:0A 26 F6 30 0C 65 F5 85<BF>
1FD8:F5 A8 8A 65 F6 85 F6 10<F3>
1FE0:17 4C 2B E5 A9 00 85 D6<7C>
1FE8:85 D7 A4 F5 84 D8 A5 F6<25>
1FF0:85 D9 D0 04 C0 00 F0 E9<30>
1FF8:A2 8E 20 88 25 38 A5 97<09>
2000:E5 8C 85 D4 A5 98 E5 8D<41>
2008:85 D5 20 C0 E5 A9 00 A8<61>
2010:A6 F6 F0 0A 91 97 C8 D0<9F>
2018:FB E6 98 CA D0 F6 A6 F5<85>
2020:F0 06 91 97 C8 CA D0 FA<54>
2028:4C A0 C8 20 C5 C9 A5 8A<47>
2030:85 BE A5 8B 85 BF A5 A1<9A>
2038:AA 0A A8 B1 80 85 8B C8<CB>
2040:B1 80 85 8A A0 01 8A D1<8F>
2048:8A D0 13 88 B1 8A C5 A0<83>
2050:B0 13 A0 02 B1 8A 65 8A<C8>
2058:90 E8 E6 8B B0 E4 90 03<77>
2060:18 90 EF 38 60 D0 FC 18<34>
2068:60 A9 C1 2C A9 C2 AA C8<67>
2070:B1 99 D0 03 C8 B1 99 49<1E>
2078:80 20 ED E5 8A D1 9D F0<87>
2080:33 91 9D A5 99 A0 02 91<88>
2088:9D C8 A5 9A 91 9D 4C E8<CE>
2090:C9 A5 99 48 A5 9A 48 A9<B9>
2098:00 A8 91 80 C8 C8 D0 FA<F5>
20A0:A5 89 85 9A A5 88 85 99<7A>
20A8:A0 04 B1 99 C9 4F F0 B9<79>
20B0:C9 5D F0 B8 A0 01 B1 99<21>
20B8:0A B0 1B A8 B1 80 D0 09<F7>
20C0:A5 9A 91 80 C8 A5 99 91<18>
20C8:80 18 A0 02 B1 99 65 99<B6>
20D0:90 D4 E6 9A B0 D0 A5 88<33>
20D8:85 99 A5 89 85 9A A0 00<18>
20E0:B1 80 D0 0C A5 9A 91 80<C4>
20E8:C8 A5 99 91 80 4C 2B CA<E7>
20F0:85 9A C8 B1 80 85 99 C8<78>
20F8:D0 E6 68 85 9A 68 85 99<EB>
2100:60 A9 1E 4C DC F8 B1 8A<AF>
2108:C8 C4 A7 B0 1A C9 1C D0<FB>
2110:1A E6 A8 20 4F E4 A5 D2<ED>
2118:C9 C2 D0 E5 A0 00 B1 D4<AB>
2120:AA C8 B1 D4 A8 8A 60 A9<69>
2128:00 A8 60 4C 13 E5 20 3A<AB>
2130.CA C0 00 30 09 85 B7 84<96>
2138:B8 A9 00 85 B6 60 4C 28<CC>
2140:E5 A5 B7 85 A0 A5 B8 85<51>
2148:A1 20 62 C9 A5 8A 85 F3<87>
2150:A5 8B 85 F4 A5 BE 85 8A<A9>
2158:A5 BF 85 8B A0 00 B1 F3<A7>
2160:85 B7 C8 B1 F3 85 B8 C8<F8>
2168:B1 F3 85 F5 84 F2 A4 F2<11>
2170:C8 B1 F3 85 F6 C8 84 F2<43>
2178:B1 F3 C9 01 F0 23 A4 F6<42>
2180:C4 F5 B0 03 88 90 E9 88<08>
2188:84 F2 A0 01 B1 F3 30 42<20>
2190:38 A5 F2 65 F3 85 F3 A9<E9>
2198:00 85 B6 65 F4 85 F4 90<76>
21A0:BB A9 00 85 F5 A5 F5 C5<DC>
21A8:B6 B0 12 E6 F2 A4 F2 B1<45>
21B0:F3 C9 9B F0 D5 C9 2C D0<EB>
21B8:F2 E6 F5 D0 E8 A9 40 85<F4>
21C0:A6 E6 F2 4C 42 CB A4 F2<38>
21C8:B1 F3 C9 2C 18 F0 02 C9<CB>
21D0:9B 60 A9 06 4C DC F8 A9<D8>
21D8:3F 85 C2 B1 8A C9 0F D0<8B>
21E0:12 20 86 E4 20 DD CB A4<D0>
21E8:A8 E6 A8 B1 8A C9 12 D0<5F>
21F0:02 66 B4 20 4F E4 C6 A8<42>
21F8:90 05 20 78 DD 85 B4 20<53>
2200:AB 2B 20 7E C2 20 5B CB<58>
2208:A0 00 84 A6 84 F2 20 4F<5A>
2210:E4 E6 A8 A5 D2 30 23 20<04>
2218:C8 2B B0 15 20 FA CA D0<9E>
2220:10 20 C0 E5 4C AC CB A5<0A>
2228:11 F0 01 60 C6 11 4C A4<C4>
2230:F5 A9 00 85 B4 A9 08 4C<05>
2238:DC F8 A0 00 A9 11 8D A2<C3>
2240:20 84 A9 84 B0 84 AA 84<6B>
2248:B1 20 3A E5 C6 F2 A5 F2<34>
2250:85 F5 A2 FF E8 E6 F2 A4<1F>
2258:F2 B1 F3 C9 9B F0 08 C9<2A>
2260:2C D0 F1 24 A6 50 ED A4<FB>
2268:F5 A5 A8 48 8A A2 F3 20<AA>
2270:8B E4 68 85 A8 20 1F C8<84>
2278:24 A6 50 11 E6 B6 A6 A8<6C>
2280:E8 E4 A7 B0 0F 20 FA CA<38>
2288:90 1A 4C BE CA A6 A8 E8<84>
2290:E4 A7 90 08 20 AB 2B A9<D2>
2298:00 85 B4 60 20 FA CA 90<C7>
22A0:03 4C 33 CB E6 F2 4C 42<5C>
22A8:CB A2 00 A5 D4 9D 44 03<E1>
22B0:A5 D5 9D 45 03 A5 D6 9D<9B>
22B8:48 03 A5 D7 9D 49 03 A9<3C>
22C0:0B 4C FE C5 00 D8 F9 DF<DA>
22C8:20 05 E5 85 A2 84 A3 4C<F6>
22D0:5D 26 20 05 E5 85 A2 84<A9>
22D8:A3 4C DE 25 20 9B E5 20<44>
22E0:9D 2C 4C 36 E5 20 9B E5<8E>
22E8:20 97 2C 4C 36 E5 20 9B<4C>
22F0:E5 20 9E 26 4C 36 E5 20<E9>
22F8:9B E5 20 00 28 4C 36 E5<82>
2300:A6 AA BD 40 21 F0 05 49<53>
2308:80 9D 40 21 60 20 DC D8<9A>
2310:90 62 F0 60 B0 50 20 DC<95>
2318:D8 F0 4B D0 57 20 DC D8<77>
2320:90 52 B0 42 20 DC D8 90<6C>
2328:3D F0 3B B0 47 20 DC D8<35>
2330:90 34 B0 40 20 DC D8 F0<15>
2338:3B D0 2B C6 AA A6 AA C6<56>
2340:AA BD 40 21 3D 41 21 0A<5C>
2348:F0 1C D0 28 C6 AA A6 AA<D8>
2350:C6 AA BD 40 21 1D 41 21<2F>
2358:0A F0 0B D0 17 A6 AA C6<56>
2360:AA BD 40 21 F0 0E A9 00<4E>
2368:A8 F0 0D A0 03 2C A0 02<54>
2370:A9 40 D0 04 A9 40 A0 01<DB>
2378:E6 AA A6 AA 9D 40 21 98<3B>
2380:9D 60 21 A9 00 9D 00 21<7F>
2388:9D 80 21 9D A0 21 9D C0<97>
2390:21 9D E0 21 60 A6 AA C6<51>
2398:AA BD 40 21 0A F0 C7 A9<43>
23A0:80 6A D0 D2 A4 A9 B9 A3<16>
23A8:20 C9 2F 90 03 4C 2E D9<8E>
23B0:A4 AA C6 AA A6 AA C6 AA<D0>
23B8:BD 40 21 D9 40 21 D0 2C<79>
23C0:0A 90 02 E8 88 BD 60 21<FE>
23C8:D9 60 21 D0 1E BD 80 21<FB>
23D0:D9 80 21 D0 16 BD A0 21<CB>
23D8:D9 A0 21 D0 0E BD C0 21<DB>
23E0:D9 C0 21 D0 06 BD E0 21<A7>
23E8:D9 E0 21 60 19 40 21 10<D8>
23F0:FA 6A 49 80 2A 60 20 DC<A5>
23F8:E4 20 9C 2E 20 DC E4 A0<66>
2400:00 A5 D6 D0 06 A5 D7 F0<CB>
2408:05 C6 D7 C6 D6 AA 08 A5<A6>
2410:E2 D0 06 A5 E3 F0 20 C6<6D>
2418:E3 C6 E2 28 F0 16 B1 D4<1F>
2420:D1 E0 D0 0E E6 D4 D0 02<C5>
2428:E6 D5 E6 E0 D0 D3 E6 E1<C9>
2430:D0 CF B0 02 C8 18 60 28<B1>
2438:38 60 20 DC E4 A5 D6 A4<79>
2440:D7 85 D4 84 D5 20 DC 2B<6A>
2448:A9 00 85 D2 85 D3 4C 3A<7D>
2450:E5 20 16 E5 A0 00 B1 D4<56>
2458:4C 79 D9 C6 AA 38 AD E5<19>
2460:02 E5 90 AA AD E6 02 E5<85>
2468:91 A8 8A 4C 79 D9 20 73<5B>
2470:C5 A9 00 85 F2 20 C8 2B<68>
2478:20 93 C5 90 CB 4C D7 F8<74>
2480:20 DC E4 A0 00 B1 D4 4C<FC>
2488:79 D9 20 73 C5 A0 00 84<AB>
2490:F2 84 D4 84 D5 20 68 EA<87>
2498:90 FB 20 93 C5 4C 7D D9<BD>
24A0:20 DC E4 4C 7D D9 A9 00<C3>
24A8:F0 0A A9 08 D0 06 A9 0C<FC>
24B0:D0 02 A9 14 85 EC 20 16<01>
24B8:E5 D0 0F C9 08 B0 0B 65<CF>
24C0:EC AA BD 70 02 A0 00 4C<62>
24C8:79 D9 4C 28 E5 20 AB 2B<37>
24D0:A0 00 A5 D5 F0 03 20 13<47>
24D8:DA A5 D4 48 4A 4A 4A 4A<CB>
24E0:20 1E DA 68 29 0F 09 30<39>
24E8:C9 3A 90 02 69 06 91 F3<C6>
24F0:C8 60 20 16 E5 20 05 DA<25>
24F8:A9 80 D0 3B 20 6E E5 20<F2>
2500:BE 2B A0 FF C8 B1 F3 10<CE>
2508:FB 29 7F 91 F3 C8 A5 F3<E5>
2510:D0 25 AD FC 02 A0 00 C9<3C>
2518:C0 B0 1A A2 0E DD 93 DC<C4>
2520:F0 13 CA 10 F8 20 0D C6<81>
2528:4C 68 DA 20 16 E5 A5 D4<CC>
2530:8D C0 05 A0 01 A9 C0 A2<69>
2538:05 86 D5 85 D4 84 D6 A9<A0>
2540:00 85 D7 85 D3 A9 83 85<E4>
2548:D2 4C 3A E5 C6 AA A9 3F<C9>
2550:85 D4 A2 05 AD 0A D2 29<31>
2558:F0 C9 A0 B0 F7 85 E0 AD<02>
2560:0A D2 29 0F C9 0A B0 F7<FF>
2568:05 E0 95 D4 CA D0 E5 20<94>
2570:B8 2D 4C 80 D9 20 86 DA<91>
2578:20 26 D8 4C FF DA A6 AA<74>
2580:BD 40 21 29 7F 9D 40 21<F7>
2588:60 20 6E E5 A5 D4 29 7F<EC>
2590:38 E9 40 90 14 AA A9 00<CA>
2598:E0 05 90 02 A2 04 95 D5<41>
25A0:CA 10 FB 20 B8 2D 4C 3A<78>
25A8:E5 E6 AA 60 20 2F D8 20<A7>
25B0:6E E5 A5 D4 29 7F 38 E9<79>
25B8:40 90 49 AA A9 00 E0 04<DA>
25C0:B0 3D 95 D6 EB 90 F7 20<31>
25C8:6E E5 A5 D4 29 7F 38 E9<13>
25D0:40 90 2F AA E0 04 B0 D1<CD>
25D8:A9 00 A8 15 D6 94 D6 E8<CF>
25E0:E0 04 D0 F7 24 D4 10 17<2C>
25E8:A8 F0 14 A9 C0 85 E0 A0<92>
25F0:01 84 E1 88 84 E2 84 E3<C9>
25F8:84 E4 84 E5 20 9D 2C 4C<17>
2600:3A E5 06 D4 20 9C 2B 90<F6>
2608:F6 B0 E0 20 6E E5 20 80<40>
2610:30 4C 38 E5 20 6E E5 20<41>
2618:8A 30 4C 38 E5 20 6E E5<99>
2620:20 48 31 4C 38 E5 20 6E<3C>
2628:E5 20 EE 2F 4C 38 E5 20<C4>
2630:6E E5 20 F2 2F 4C 38 E5<97>
2638:20 6E E5 20 44 2F 4C 38<74>
2640:E5 20 6E E5 20 C5 31 4C<29>
2648:38 E5 4C 9E D8 4C AC D8<7C>
2650:A5 E0 10 F6 4C 28 E5 20<B5>
2658:9B E5 A5 E0 F0 EF A5 D4<A6>
2660:F0 EE 10 26 29 7F 85 D4<A8>
2668:A5 E0 29 7F 38 E9 40 90<B0>
2670:E3 A2 04 C9 04 F0 0E B0<4F>
2678:11 AA A8 B9 E2 00 D0 D4<03>
2680:C8 C0 04 D0 F6 B5 E1 4A<BC>
2688:B0 01 18 A9 00 6A 48 A5<55>
2690:E0 C9 40 D0 4D A5 E2 05<E9>
2698:E3 05 E4 05 E5 D0 43 A5<F6>
26A0:E1 29 F0 4A 85 FC 4A 4A<44>
26A8:65 FC 85 FC A5 E1 29 0F<A0>
26B0:65 FC 85 FC 20 B5 2E 46<14>
26B8:FC B0 03 20 BB 2F 20 D4<2F>
26C0:2E A5 FC F0 50 20 F3 2E<E1>
26C8:20 9B 26 B0 0F 46 FC 90<A0>
```

Listing »Turbo-Basic XL« (Fortsetzung)

```
26D0:F7 20 B5 2E 20 23 2F 20<25>
26D8:9E 26 90 E2 4C 2E E5 4C<3B>
26E0:28 E5 A5 E5 48 A5 E4 48<81>
26E8:A5 E3 48 A5 E2 48 A5 E1<0D>
26F0:48 A5 E0 48 20 F2 2F B0<E1>
26F8:E6 68 85 E0 68 85 E1 68<26>
2700:85 E2 68 85 E3 68 85 E4<14>
2708:68 85 E5 20 9E 26 B0 CC<C2>
2710:20 59 2F B0 C7 68 10 04<5D>
2718:05 D4 85 D4 4C 3A E5 A9<69>
2720:06 2C A9 00 85 FB 60 20<88>
2728:16 E5 85 E0 84 E1 4C 16<5F>
2730:E5 20 5F DC 98 25 E1 A8<45>
2738:A5 E0 25 D4 4C 79 D9 20<4D>
2740:5F DC 98 05 E1 A8 A5 E0<51>
2748:05 D4 4C 79 D9 20 5F DC<11>
2750:98 45 E1 A8 A5 E0 45 D4<DC>
2758:4C 79 D9 9A 98 9D 9B B3<0B>
2760:B5 B0 B2 A6 3C 7C BC 27<34>
2768:67 A7 20 13 E5 A5 14 A4<D3>
2770:11 F0 11 C5 14 F0 F8 A5<A7>
2778:D4 D0 04 C6 D5 30 05 C6<B8>
2780:D4 4C A5 DC 60 A6 AA CA<A0>
2788:E6 AA E6 AA 18 BD 00 21<B4>
2790:9D 02 21 BD 01 21 9D 03<78>
2798:21 8A 69 20 AA 90 EE 60<B8>
27A0:20 BD DC 20 E4 DA 20 26<98>
27A8:D8 4C 1D D8 A5 C3 4C FD<E0>
27B0:D9 A5 BA A4 BB 4C 79 D9<D4>
27B8:E6 C9 A5 C9 F0 FA 85 AF<FE>
27C0:A9 00 85 94 A4 A8 B1 8A<85>
27C8:C9 12 F0 4C C9 16 F0 7A<2A>
27D0:C9 14 F0 76 C9 15 F0 6D<D4>
27D8:C9 1C F0 5A 20 00 E4 C6<B5>
27E0:A8 A6 AA BD 00 21 30 03<A4>
27E8:20 34 DA 20 DC E4 A6 B5<F7>
27F0:20 AF C2 A5 D4 9D 44 03<67>
27F8:A5 D5 9D 45 03 A5 D6 9D<A9>
2800:48 03 18 65 94 85 94 A5<09>
2808:D7 9D 49 03 05 D6 F0 B4<99>
2810:A9 0B 20 FE C5 4C FC DC<86>
2818:A4 94 C8 C4 AF 90 09 18<32>
2820:A5 C9 65 AF 85 AF 90 F0<AE>
2828:A4 94 C4 AF B0 17 20 02<EE>
2830:F2 E6 94 4C 60 DD 20 78<D9>
2838:DD 85 B5 C6 A8 4C FC DC<CD>
2840:E6 A8 4C 13 E5 E6 A8 4C<CA>
2848:FC DC A4 A8 88 B1 8A C9<6B>
2850:15 F0 07 C9 12 F0 03 20<0A>
2858:94 C2 A9 00 85 B5 60 A9<03>
2860:13 85 F3 A9 23 85 F4 C6<21>
2868:DB A2 07 86 B5 A9 00 A0<B0>
2870:08 20 B4 C3 20 B9 C4 20<06>
2878:F2 DC 4C F8 C4 A9 5F 2C<91>
2880:A9 FF 85 DF A4 B0 88 98<0A>
2888:F0 03 20 16 E5 85 DA 84<69>
2890:DB 20 DC E4 20 9C 2E 20<FF>
2898:DC E4 18 A5 D4 65 DA 85<96>
28A0:D4 A5 D5 65 DB 85 D5 B0<47>
28A8:22 38 A5 D6 E5 DA 85 D6<F9>
28B0:A5 D7 E5 DB 85 D7 90 13<8E>
28B8:05 D6 F0 0F 38 A5 D6 E5<F0>
28C0:E2 85 DC A5 D7 E5 E3 85<E5>
28C8:DD B0 03 4C 9E D8 E6 DA<CB>
28D0:D0 02 E6 DB A0 00 B1 E0<39>
28D8:51 D4 25 DF D0 30 A5 D4<BC>
28E0:85 D8 A5 D5 85 D9 A5 E0<A5>
28E8:85 E4 A5 E1 85 E5 A5 E3<7C>
28F0:85 DE A6 E2 E8 CA D0 04<1B>
28F8:C6 DE 30 25 B1 D8 51 E4<68>
2900:25 DF D0 0A C8 D0 EE E6<76>
2908:D9 E6 E5 4C 2D DE E6 D4<32>
2910:D0 02 E6 D5 A5 DC D0 04<0D>
2918:C6 DD 30 AF C6 DC 4C 06<B1>
2920:DE A5 DA A4 DB 4C 79 D9<D7>
2928:85 D6 85 D7 A0 08 16 E6<67>
2930:36 E5 26 D6 90 0B 18 B5<8E>
2938:E6 65 D7 95 E6 90 02 F6<B2>
2940:E5 88 D0 EA 60 4C 28 E5<2F>
2948:20 05 E5 D0 F8 48 A4 A8<57>
2950:C8 C4 A7 B0 05 20 13 E5<37>
2958:D0 EB 68 D0 02 A9 01 85<91>
2960:D5 A2 16 A9 00 95 D6 CA<9A>
2968:10 FB A5 D5 85 E0 E8 20<1E>
2970:60 DE A5 D4 A2 02 20 60<11>
2978:DE A5 D5 85 D6 A0 08 06<2D>
2980:DC 26 DB 26 DA 06 D6 90<32>
2988:11 18 A5 DC 65 E8 85 DC<8D>
2990:A5 DB 65 E7 85 DB 90 02<B9>
2998:E6 DA 88 D0 E2 20 37 DF<1D>
29A0:20 5C DF 20 37 DF 20 6A<4E>
29A8:DF 20 4E DF 20 5C DF 20<76>
29B0:4E DF 20 6A DF 24 E2 30<06>
29B8:21 E6 E1 18 A5 EB 65 E6<3B>
29C0:85 EB A5 EA 65 E5 85 EA<74>
29C8:90 02 E6 E9 38 A2 02 B5<4B>
29D0:E2 F5 E9 95 E2 CA 10 F7<04>
29D8:30 C3 A5 E0 F0 2F C6 E0<6F>
29E0:38 A5 DC E5 E8 85 DC A5<33>
29E8:DB E5 E7 85 DB B0 02 C6<89>
29F0:DA 18 A2 02 B5 E2 75 DA<83>
29F8:95 E2 CA 10 F7 30 9E 18<12>
2A00:A5 99 65 E0 85 55 A5 9A<D0>
2A08:69 00 85 56 60 A2 00 86<8E>
2A10:54 86 55 86 56 60 38 A5<FC>
2A18:99 E5 E0 85 55 A5 9A E9<8B>
2A20:00 85 56 60 18 A5 9B 65<33>
2A28:E1 85 54 A5 9C 69 00 F0<59>
2A30:0E 60 38 A5 9B E5 E1 85<A4>
2A38:54 A5 9C E9 00 D0 F2 A4<A9>
2A40:C8 A2 60 20 DE 24 C0 B0<EE>
2A48:D0 E7 4C A9 C2 84 96 A2<61>
2A50:00 86 AF 85 95 A6 F2 A0<07>
2A58:00 B1 95 F0 27 BD 80 05<9C>
2A60:29 7F E8 51 95 D0 03 C8<F5>
2A68:D0 F3 0A F0 15 B0 06 C8<DD>
2A70:B1 95 10 FB 38 E6 AF F0<9A>
2A78:0D 98 65 95 90 D5 E6 96<90>
2A80:B0 D1 18 60 38 60 A9 04<7E>
2A88:4C DC F8 A2 00 86 AF 84<CA>
2A90:96 85 95 A6 F2 A0 01 B1<92>
2A98:95 F0 E9 BD 80 05 29 7F<0D>
2AA0:E8 C9 2E F0 DD 51 95 D0<F5>
2AA8:03 C8 D0 EF 0A F0 D3 B0<BF>
2AB0:05 C8 B1 95 10 FB E6 AF<FF>
2AB8:38 98 65 95 90 D3 E6 96<31>
2AC0:B0 CF 00 E4 F1 FF A0 00<31>
2AC8:A9 11 8D A2 20 84 A9 84<C9>
2AD0:B0 84 AA 84 B1 20 4F E4<63>
2AD8:B0 06 20 3A E5 4C 0F E4<87>
2AE0:85 AB AA BD CC 23 85 AC<62>
2AE8:A4 A9 BE A2 20 BD 2A 24<C6>
2AF0:C5 AC 90 0E AA F0 16 B9<56>
2AF8:A2 20 C6 A9 20 48 E4 4C<BE>
2B00:22 E4 A5 AB C8 99 A2 20<D5>
2B08:84 A9 4C 0F E4 60 0A 8D<02>
2B10:4D E4 6C 2F 23 A4 A8 E6<E2>
2B18:A8 B1 8A 30 5A F0 53 C9<F3>
2B20:0F 90 03 F0 27 60 C8 B1<3C>
2B28:8A 85 D4 C8 B1 8A 85 D5<49>
2B30:C8 B1 8A 85 D6 C8 B1 8A<53>
2B38:85 D7 C8 B1 8A 85 D8 C8<F9>
2B40:B1 8A 85 D9 C8 84 A8 A9<46>
2B48:00 85 D2 60 C8 B1 8A A2<EE>
2B50:8A 85 D6 85 D8 C8 98.18<3D>
2B58:75 00 85 D4 A9 00 85 D7<7E>
2B60:85 D9 75 01 85 D5 98 65<7A>
2B68:D6 A8 A9 83 85 D2 84 A8<F5>
2B70:18 60 C8 E6 A8 B1 8A 49<A8>
2B78:80 85 D3 20 ED E5 B1 9D<45>
2B80:85 D2 A0 02 B1 9D 85 D4<7E>
2B88:C8 B1 9D 85 D5 C8 B1 9D<DF>
2B90:85 D6 C8 B1 9D 85 D7 C8<17>
2B98:B1 9D 85 D8 C8 B1 9D 85<9C>
2BA0:D9 60 20 6E E5 A9 02 24<48>
2BA8:D2 D0 15 05 D2 85 D2 4A<61>
2BB0:90 3F 18 A5 D4 65 8C 85<2B>
2BB8:D4 A8 A5 D5 65 8D 85 D5<9F>
2BC0:60 20 13 E5 10 FA A9 07<CF>
2BC8:4C DC F8 20 13 E5 85 99<4C>
2BD0:84 9A 20 13 E5 85 9B 84<F6>
2BD8:9C 20 00 E4 20 6E E5 20<23>
2BE0:25 33 B0 0A A5 D4 A4 D5<BE>
2BE8:60 20 13 E5 F0 D2 A9 03<42>
2BF0:2C A9 09 2C A9 0B 2C A9<E3>
2BF8:0A 4C DC F8 B0 F6 B0 EE<0C>
2C00:E6 AA A4 AA C0 20 B0 EF<D1>
2C08:A5 D9 99 E0 21 A5 D8 99<FE>
2C10:C0 21 A5 D7 99 A0 21 A5<C3>
2C18:D6 99 80 21 A5 D5 99 60<AF>
2C20:21 A5 D4 99 40 21 A5 D3<96>
2C28:99 20 21 A5 D2 99 00 21<D5>
2C30:60 20 00 E4 A4 AA C6 AA<C8>
2C38:B9 E0 21 85 D9 B9 C0 21<1F>
2C40:85 D8 B9 A0 21 85 D7 B9<E4>
2C48:80 21 85 D6 B9 60 21 85<45>
2C50:D5 B9 40 21 85 D4 B9 20<68>
2C58:21 85 D3 B9 00 21 85 D2<08>
2C60:60 C6 AA A4 AA B9 E1 21<E6>
2C68:85 E5 B9 C1 21 85 E4 B9<47>
2C70:A1 21 85 E3 B9 81 21 85<D7>
2C78:E2 B9 61 21 85 E1 B9 41<CA>
2C80:21 85 E0 4C 70 E5 A5 D3<80>
2C88:20 ED E5 A5 D2 91 9D C8<E9>
2C90:A5 D3 91 9D C8 A5 D4 91<E5>
2C98:9D C8 A5 D5 91 9D C8 A5<48>
2CA0:D6 91 9D C8 A5 D7 91 9D<80>
2CA8:C8 A5 D8 91 9D C8 A5 D9<BF>
2CB0:91 9D 60 0A 2A 2A 2A A8<BB>
2CB8:6A 29 F8 18 65 86 85 9D<E7>
2CC0:98 29 07 65 87 85 9E A0<F9>
2CC8:00 60 A9 00 85 92 85 CA<8E>
2CD0:AD E7 02 AC E8 02 85 80<84>
2CD8:84 81 C8 85 82 84 83 85<7F>
2CE0:84 84 85 18 69 01 90 01<D6>
2CE8:C8 85 86 84 87 85 88 84<39>
2CF0:89 85 8A 84 8B 18 69 03<FD>
2CF8:90 01 C8 85 8C 84 8D 85<BC>
2D00:8E 84 8F 85 90 84 91 85<82>
2D08:0E 84 0F A9 00 A8 91 84<FA>
2D10:91 8A C8 A9 80 91 8A C8<BF>
2D18:A9 03 91 8A A9 0A 85 C9<99>
2D20:4C 66 FF A2 FF 9A D8 A5<68>
2D28:CA F0 03 20 04 E6 20 E7<23>
2D30:F6 20 34 C5 20 D1 F5 A5<33>
2D38:92 F0 03 20 93 C5 20 5C<84>
2D40:C5 20 CB C0 A5 CA D0 E3<91>
2D48:A2 FF 9A D8 20 AB 2B A9<0F>
2D50:5D 85 C2 20 87 C2 A5 11<8C>
2D58:D0 04 C6 11 D0 E6 A0 00<BA>
2D60:84 F2 84 9F 84 94 84 A6<F5>
2D68:84 B3 84 B0 84 B1 A5 84<9D>
2D70:85 AD A5 85 85 AE 20 1D<8D>
2D78:EB 20 FB E7 20 F7 E8 A5<6F>
2D80:D5 10 02 85 A6 20 1D EB<AC>
2D88:84 A8 B1 F3 C9 9B D0 07<2B>
2D90:24 A6 30 B0 4C E0 E7 A5<B4>
2D98:94 85 A7 20 F7 E8 20 1D<69>
2DA0:EB A0 EB A9 5A 20 C3 DF<F8>
2DA8:66 EC 30 08 A5 AF C9 15<3D>
2DB0:D0 23 86 DA A0 EC A9 80<11>
2DB8:20 C3 DF A9 36 B0 06 65<1A>
2DC0:AF 69 02 90 10 24 EC 30<5E>
2DC8:0C A9 BF 85 95 A9 EB 85<1C>
2DD0:96 A9 15 A6 DA 86 F2 20<75>
2DD8:F7 E8 20 B4 2B 20 21 E8<E2>
2DE0:90 35 A4 9F B1 F3 C9 9B<3B>
2DE8:D0 06 C8 91 F3 88 A9 20<E8>
2DF0:09 80 91 F3 A9 40 05 A6<2E>
2DF8:85 A6 A4 A8 84 F2 A2 03<67>
2E00:86 A7 E8 86 94 A9 37 20<FB>
2E08:F7 E8 A4 F2 B1 F3 E6 F2<7C>
2E10:C9 9B D0 F3 20 F7 E8 A5<32>
2E18:94 A4 A7 99 00 21 A4 F2<65>
2E20:88 B1 F3 C9 9B F0 03 4C<7E>
2E28:D1 E6 A0 02 A5 94 99 00<C5>
2E30:21 20 5F C9 A9 00 B0 04<56>
2E38:A0 02 B1 8A 38 E5 94 F0<A0>
2E40:24 B0 13 49 FF A8 C8 A2<4B>
2E48:8A 20 86 25 A5 97 85 8A<09>
2E50:A5 98 85 8B D0 0F A8 18<F0>
2E58:65 8A 85 8A 90 02 E6 8B<F6>
2E60:A2 8A 20 20 26 A4 94 88<A5>
2E68:B9 00 21 91 8A 98 D0 F7<9F>
2E70:24 A6 50 2A A5 B1 0A 0A<FA>
2E78:0A A8 A2 88 20 20 26 38<96>
2E80:A5 84 E5 AD A8 A5 85 E5<39>
2E88:AE A2 84 20 22 26 24 A6<4A>
2E90:10 06 20 31 F2 4C 7E E6<31>
2E98:20 07 F2 4C 7E E6 10 FB<EF>
2EA0:20 26 C1 4C AF FF 20 5F<F3>
2EA8:C9 B0 F0 A0 02 B1 8A A8<6E>
2EB0:18 65 8A 85 8A 90 02 E6<77>
2EB8:8B A2 8A 20 20 26 4C 7E<45>
2EC0:E6 20 C8 2B 90 08 A9 00<B8>
2EC8:85 F2 A0 80 30 0B 20 25<8A>
2ED0:33 B0 F3 A4 D5 30 EF A5<86>
2ED8:D4 84 A1 85 A0 20 F7 E8<FF>
2EE0:A5 A1 85 D5 4C F7 E8 A0<82>
2EE8:00 84 A9 B1 95 0A A8 B9<7C>
2EF0:39 F0 85 9D 8D 82 04 B9<9A>
2EF8:3A F0 85 9E 8D 83 04 A5<17>
2F00:94 8D 81 04 A5 F2 8D 80<43>
2F08:04 E6 9D D0 02 E6 9E A2<19>
2F10:00 A1 9D 30 0C C9 05 90<4A>
2F18:4F 20 E6 E8 90 EB 4C BC<65>
2F20:E8 0A A8 B9 3A F0 48 B9<50>
2F28:39 F0 48 C0 12 B0 0F 68<DD>
2F30:A8 68 20 74 E8 90 D2 4C<9D>
2F38:BC E8 48 98 48 60 A6 A9<25>
2F40:E8 E8 E8 E8 F0 1F 86 A9<BE>
2F48:A5 F2 9D 80 04 A5 94 9D<FA>
2F50:81 04 A5 9D 9D 82 04 A5<76>
2F58:9E 9D 83 04 68 85 9D 68<90>
2F60:85 9E 4C 43 E8 4C 00 E9<BF>
2F68:A6 A9 D0 01 60 BD 82 04<E6>
2F70:85 9D BD 83 04 85 9E CA<85>
2F78:CA CA CA 86 A9 B0 03 4C<29>
2F80:43 E8 E6 9D D0 02 E6 9E<B9>
2F88:A2 00 A1 9D 30 F4 C9 03<7B>
2F90:F0 D6 B0 EE A5 F2 C5 9F<50>
2F98:90 02 85 9F A6 A9 BD 80<8B>
```

```
2FA0:04 85 F2 BD 81 04 85 94<AF>
2FA8:4C 43 E8 60 C9 0F D0 41<63>
2FB0:E6 9D D0 02 E6 9E A2 00<6D>
2FB8:A1 9D 18 C6 94 A4 94 99<6A>
2FC0:00 21 E6 94 D0 E5 A9 0E<DF>
2FC8:4C DC F8 A2 FF 9A A5 94<AC>
2FD0:A4 A7 99 00 21 4C D1 E6<DE>
2FD8:A2 FF 9A A0 04 B9 00 21<3C>
2FE0:C9 54 D0 0A A4 F2 88 A9<F7>
2FE8:9B 91 F3 4C 51 E7 4C 44<1D>
2FF0:E7 20 1D EB C4 B3 F0 14<F8>
2FF8:84 B3 A0 ED A9 71 20 85<77>
3000:DF B0 2C 86 B2 18 A5 AF<B8>
3008:69 10 85 B0 A0 00 B1 9D<57>
3010:C5 B0 F0 12 C9 44 D0 0A<1F>
3018:A5 B0 C9 56 B0 04 C9 44<D8>
3020:B0 02 38 60 A5 B0 20 F7<61>
3028:E8 A6 B2 86 F2 18 60 A9<6E>
3030:00 85 B0 38 60 A9 C0 D0<60>
3038:06 A9 00 F0 02 A9 80 85<FF>
3040:D2 20 1D EB 84 AC 20 53<45>
3048:EA B0 28 20 2B E9 A5 B0<76>
3050:F0 08 A4 B2 B1 F3 C9 30<75>
3058:90 19 E6 F2 20 53 EA 90<50>
3060:F9 C9 30 90 04 C9 3A 90<C4>
3068:F1 C9 24 F0 08 24 D2 10<B9>
3070:0D 70 18 38 60 24 D2 10<41>
3078:FA 70 F8 C8 D0 0D B1 F3<82>
3080:C9 28 D0 07 C8 A9 40 05<28>
3088:D2 85 D2 A5 AC 85 F2 84<93>
3090:AC A4 83 A5 82 20 85 DF<17>
3098:B0 0A E4 AC F0 41 20 AC<FB>
30A0:DF 4C D2 E9 38 A5 AC E5<3E>
30A8:F2 85 F2 A8 A2 84 20 86<10>
30B0:25 A5 AF 85 D3 A4 F2 88<8F>
30B8:A6 AC CA BD 80 05 09 80<05>
30C0:91 97 CA BD 80 05 88 10<CA>
30C8:F7 A0 08 A2 88 20 86 25<93>
30D0:E6 B1 20 9C 2B A0 07 B9<BC>
30D8:D2 00 91 97 88 10 F8 98<D1>
30E0:48 A5 9D 48 A6 9E A5 AF<41>
30E8:20 ED E5 B1 9D 45 D2 A8<90>
30F0:86 9E 68 85 9D C0 80 68<4E>
30F8:A8 B0 A3 24 D2 50 04 30<B6>
3100:02 C6 AC A5 AC 85 F2 A5<8E>
3108:AF 10 07 A9 00 20 F7 E8<87>
3110:A5 AF 49 80 20 F7 E8 18<96>
3118:60 A4 F2 B1 F3 C9 5F F0<D9>
3120:F6 C9 41 90 07 C9 5B 60<C3>
3128:A4 AC 84 F2 38 60 A4 F2<ED>
3130:B1 F3 38 E9 30 90 F5 C9<30>
3138:0A 90 0A C9 11 90 ED E9<DC>
3140:07 C9 10 B0 E7 A0 04 06<1D>
3148:D4 26 D5 88 D0 F9 05 D4<79>
3150:85 D4 E6 F2 18 60 20 1D<E1>
3158:EB 84 AC B1 F3 C9 24 F0<84>
3160:07 20 C8 2B 90 20 B0 C0<1E>
3168:E6 F2 20 1D EB 20 9C 2B<42>
3170:20 68 EA B0 B3 20 68 EA<04>
3178:B0 06 A5 D5 29 F0 F0 F5<F3>
3180:20 DC 2B A9 0D 2C A9 0E<52>
3188:20 F7 E8 A4 94 A2 00 B5<DC>
3190:D4 99 00 21 C8 E8 E0 06<B8>
3198:90 F5 84 94 18 60 20 1D<75>
31A0:EB B1 F3 C9 22 F0 02 38<27>
31A8:60 A9 0F 20 F7 E8 A5 94<80>
31B0:85 AB 20 F7 E8 E6 F2 A4<71>
31B8:F2 B1 F3 C9 9B F0 0D C9<EA>
31C0:22 D0 EF E6 F2 C8 B1 F3<68>
31C8:C9 22 F0 E6 18 A5 94 E5<D8>
31D0:AB A4 AB 99 00 21 18 60<B0>
31D8:AD 01 21 0A B0 04 A5 94<00>
31E0:C9 06 60 A4 F2 B1 F3 29<DF>
31E8:7F C9 20 D0 03 C8 D0 F5<CA>
31F0:84 F2 B1 F3 C9 9B F0 25<8F>
31F8:29 7F 91 F3 C9 61 90 08<D2>
3200:C9 7B B0 04 29 5F 91 F3<D6>
3208:C8 C9 30 90 10 C9 3A 90<29>
3210:E1 C9 41 90 08 C9 5B 90<E4>
3218:D9 C9 5F F0 D5 A4 F2 60<63>
3220:95 52 45 CD 95 44 41 54<ED>
3228:C1 98 49 4E 50 55 D4 91<5D>
3230:43 4F 4C 4F D2 9D 4C 49<32>
3238:53 D4 96 45 4E 54 45 D2<92>
3240:9A 4C 45 D4 9B 49 C6 9C<D0>
3248:46 4F D2 97 4E 45 58 D4<6B>
3250:91 47 4F 54 CF 91 47 4F<D8>
3258:20 54 CF 91 47 4F 53 55<53>
3260:C2 BD 54 52 41 D0 90 42<1E>
3268:59 C5 90 43 4F 4E D4 9E<D6>
3270:43 4F CD 9F 43 4C 4F 53<D6>
3278:C5 90 43 4C D2 90 44 45<E2>
3280:C7 9E 44 49 CD 90 45 4E<88>
3288:C4 90 4E 45 D7 A0 4F 50<2C>
3290:45 CE 96 4C 4F 41 C4 96<AF>
3298:53 41 56 C5 A2 53 54 41<BF>
32A0:54 55 D3 A3 4E 4F 54 C5<FB>
32A8:CF 50 4F 49 4E D4 A1 58<9B>
32B0:49 CF A4 4F CE 92 50 4F<7E>
32B8:4B C5 A5 50 52 49 4E D4<F1>
32C0:90 52 41 C4 99 52 45 41<FE>
32C8:C4 A8 52 45 53 54 4F 52<8B>
32D0:C5 90 52 45 54 55 52 CE<9F>
32D8:A9 52 55 CE 90 53 54 4F<6E>
32E0:D0 90 50 4F D0 A5 BF 98<37>
32E8:47 45 D4 AB 50 55 D4 91<50>
32F0:47 52 41 50 48 49 43 D3<36>
32F8:92 50 4C 4F D4 92 50 4F<6C>
3300:53 49 54 49 4F CE 90 44<07>
3308:4F D3 92 44 52 41 57 54<67>
3310:CF 93 53 45 54 43 4F 4C<C7>
3318:4F D2 AC 4C 4F 43 41 54<A0>
3320:C5 94 53 4F 55 4E C4 A6<B7>
3328:4C 50 52 49 4E D4 90 43<83>
3330:53 41 56 C5 90 43 4C 4F<18>
3338:41 C4 9A 00 80 00 45 52<43>
3340:52 4F 52 2D 20 A0 92 44<99>
3348:50 4F 4B C5 93 4D 4F 56<26>
3350:C5 93 2D 4D 4F 56 C5 AA<5D>
3358:2A C6 90 52 45 50 45 41<B1>
3360:D4 91 55 4E 54 49 CC 91<42>
3368:57 48 49 4C C5 90 57 45<96>
3370:4E C4 90 45 4C 53 C5 90<FA>
3378:45 4E 44 49 C6 CF 42 50<E7>
3380:55 D4 CF 42 47 45 D4 92<9D>
3388:46 49 4C 4C 54 CF 90 44<65>
3390:CF 90 4C 4F 4F D0 A8 45<00>
3398:58 49 D4 DA 44 49 D2 96<93>
33A0:4C 4F 43 CB 96 55 4E 4C<DF>
33A8:4F 43 CB 96 52 45 4E 41<9A>
33B0:4D C5 96 44 45 4C 45 54<1A>
33B8:C5 91 50 41 55 53 C5 96<0A>
33C0:54 49 4D 45 24 BD A7 50<43>
33C8:52 4F C3 B8 45 58 45 C3<E0>
33D0:90 45 4E 44 50 52 4F C3<C2>
33D8:91 46 43 4F 4C 4F D2 AA<3E>
33E0:2A CC 95 2D AD 93 52 45<4F>
33E8:4E 55 CD 92 44 45 CC DA<F2>
33F0:44 55 4D D0 AA 54 52 41<B8>
33F8:43 C5 DB 54 45 58 D4 96<86>
3400:42 4C 4F 41 C4 96 42 52<48>
3408:55 CE B8 47 4F A3 A7 A3<70>
3410:AA 2A C2 92 50 41 49 4E<9D>
3418:D4 9F 43 4C D3 94 44 53<80>
3420:4F 55 4E C4 DC 43 49 52<1B>
3428:43 4C C5 AB 25 50 55 D4<51>
3430:98 25 47 45 D4 9A 00 82<F2>
3438:80 AC A4 BA BB 9B 47 4F<09>
3440:54 CF 47 4F 53 55 C2 54<19>
3448:CF 53 54 45 D0 54 48 45<6F>
3450:CE A3 3C BD 3C BE 3E BD<AD>
3458:BC BE BD DE AA AB AD AF<87>
3460:4E 4F D4 4F D2 41 4E C4<F7>
3468:A8 A9 BD BD 3C BD 3C BE<41>
3470:3E BD BC BE BD AB AD A8<30>
3478:80 80 A8 A8 AC 53 54 52<BD>
3480:A4 43 48 52 A4 55 53 D2<6A>
3488:41 53 C3 56 41 CC 4C 45<CA>
3490:CE 41 44 D2 41 54 CE 43<0B>
3498:4F D3 50 45 45 CB 53 49<E1>
34A0:CE 52 4E C4 46 52 C5 45<0F>
34A8:58 D0 4C 4F C7 43 4C 4F<D7>
34B0:C7 53 51 D2 53 47 CE 41<64>
34B8:42 D3 49 4E D4 50 41 44<D6>
34C0:44 4C C5 53 54 49 43 CB<44>
34C8:50 54 52 49 C7 53 54 52<C8>
34D0:49 C7 44 50 45 45 CB A6<96>
34D8:A1 49 4E 53 54 D2 49 4E<9E>
34E0:4B 45 59 A4 45 58 4F D2<7C>
34E8:48 45 58 A4 44 45 C3 44<DD>
34F0:49 D6 46 52 41 C3 54 49<55>
34F8:4D 45 A4 54 49 4D C5 4D<D3>
3500:4F C4 45 58 45 C3 52 4E<C6>
3508:C4 52 41 4E C4 54 52 55<5D>
3510:4E C3 25 B0 25 B1 25 B2<59>
3518:25 B3 47 4F A3 55 49 4E<EA>
3520:53 54 D2 45 52 D2 45 52<45>
3528:CC 00 2B 8C 2C 8B 02 89<2B>
3530:8C 02 8A 8B 03 25 0F 35<67>
3538:02 26 0F 36 02 28 03 B0<C9>
3540:02 BD 02 81 02 B4 03 88<83>
3548:8C 02 03 82 8E 03 2B 0F<D7>
3550:38 8C 8F 2C 02 03 12 0F<DD>
3558:3C 8C 02 03 58 BA 02 6B<4C>
3560:BA 02 60 02 66 02 67 02<5E>
3568:68 02 69 02 6C 02 6D 02<9A>
3570:B1 02 5E B2 02 64 B2 02<9F>
3578:65 B2 02 44 B2 02 D1 B3<F0>
3580:02 48 0F 63 03 3F D0 D3<8D>
3588:2C 03 D0 8C 2C 03 D0 B5<CE>
3590:2C 03 B5 BB B5 03 59 02<F9>
3598:5F 02 B6 02 B7 02 83 03<D8>
35A0:D2 B2 03 85 B9 03 2B 0F<B7>
35A8:37 8C 8F 2C 02 03 1D 0F<8A>
35B0:2F 02 1E 0F 30 02 1F 0F<B5>
35B8:31 02 20 0F 32 02 21 0F<0A>
35C0:33 02 22 0F 34 03 C0 C6<6C>
35C8:90 03 8C 90 03 8D 22 0F<45>
35D0:2D 8C 90 02 B7 22 0F 2E<8C>
35D8:B5 90 03 82 22 0F 2D 8C<3C>
35E0:19 8C BC 90 03 1A 8C 02<14>
35E8:03 C4 12 97 03 C4 12 AF<6E>
35F0:90 03 BE D8 82 90 03 90<24>
35F8:02 91 02 1C B8 03 D9 BF<CC>
3600:D6 90 03 14 02 16 03 BE<49>
3608:90 02 C0 CB 90 03 1C 8C<B4>
3610:03 82 02 85 03 BF D6 02<A7>
3618:03 12 D5 02 03 8C 12 BE<71>
3620:12 C4 12 B5 90 03 BE 90<A8>
3628:02 90 03 96 02 90 03 BE<85>
3630:D8 02 03 BE D8 02 83 D8<EB>
3638:02 03 C1 91 02 C1 92 02<E1>
3640:C1 8C 12 90 02 96 02 90<75>
3648:03 B5 12 02 03 BE 12 82<7A>
3650:12 97 03 8C 12 8C 03 90<1D>
3658:02 8C 12 8C 12 C4 90 03<A9>
3660:C8 D7 90 03 8C C5 90 03<44>
3668:86 C3 03 12 C2 02 03 17<04>
3670:C6 02 18 C6 02 62 C2 02<8A>
3678:6A C2 03 8C C7 03 12 C6<4F>
3680:02 03 82 2B 0F 39 8C 8F<A5>
3688:2C 02 85 2B 0F 3B 8C 2C<16>
3690:03 C8 D7 02 03 12 C9 02<5F>
3698:03 8C 1B CA 90 02 8C 90<DC>
36A0:03 81 02 84 AE 02 CD AD<22>
36A8:02 03 AE 02 03 AF CC 03<19>
36B0:8C 02 B5 03 CD AD 02 03<ED>
36B8:D8 CE 03 CD 02 03 12 02<F2>
36C0:15 03 40 02 41 02 42 02<2E>
36C8:43 02 5C 03 3D 02 3E 02<60>
36D0:5B 03 8C D4 03 12 0F 3C<33>
36D8:D3 02 03 2B 0F 3A 03 BE<56>
36E0:12 92 03 80 D0 B5 12 0F<DA>
36E8:3C 85 8F 2C 03 87 86 90<07>
36F0:03 26 90 02 25 90 02 90<95>
36F8:03 8C 12 93 02 93 03 11<A9>
3700:E9 8F EA 72 E9 D7 EA 04<65>
3708:E9 76 E9 6E E9 11 EB F2<9F>
3710:F0 6E EE 78 EE 80 EE 63<B3>
3718:EE 84 EE 87 EE 8F EE 3C<CA>
3720:EF 03 EF 96 EF 94 EF 90<45>
3728:EF 1C F0 5C EF 2D EF 37<19>
3730:EF 38 EF 06 EF D2 EF 14<12>
3738:EF 73 EF 99 EF 5F EF 58<8D>
3740:EF 56 EF 2B EF 86 EF 9D<54>
3748:EF 40 EF 44 EF 26 F0 30<74>
3750:EF 64 EF 2A F0 FF EE 22<08>
3758:EF E3 EF E6 EF E9 EF 95<C1>
3760:EE BE EE C3 EE C7 EE CB<A0>
3768:EE CF EE D9 EE DC EE 27<D2>
3770:F0 DF EE 1D F0 E7 EE 1E<33>
3778:EF 32 EF 47 EF 4A EF 68<23>
3780:EF 82 EF A1 EF A4 EF 8C<2E>
3788:EF A8 EF B4 EF B7 EF BB<60>
3790:EF CA EF DA EF DD EF ED<1A>
3798:EF F1 EF F4 EF 18 F0 14<23>
37A0:F0 FB EF 05 F0 0B F0 0E<8C>
37A8:F0 4E EF 52 EF CE EF F7<3C>
37B0:EF 6C EF 7E EF 26 EF 32<86>
37B8:F0 20 2B E9 A5 B0 C9 1C<B5>
37C0:90 1C C9 28 F0 18 C9 2B<B5>
37C8:90 16 C9 56 F0 12 C9 57<04>
37D0:F0 0E C9 5A F0 0A C9 5D<14>
37D8:F0 06 C9 61 F0 02 38 60<46>
37E0:4C 60 E9 20 1E F4 A0 00<6D>
37E8:84 A0 84 A1 84 B9 88 84<44>
37F0:AD A9 7F 85 AE 8D FE 02<98>
37F8:20 94 C2 A4 A8 C8 C4 A7<6A>
3800:B0 2E A5 A8 48 20 00 E4<89>
3808:68 85 A8 A5 D2 10 06 20<97>
3810:BE C2 4C 35 F1 20 FB E4<0F>
3818:84 A1 85 A0 A6 A8 E4 A7<08>
3820:F0 0A A0 80 E8 E4 A7 F0<23>
3828:03 20 FB E4 85 AD 84 AE<5D>
3830:20 62 C9 A0 01 B1 8A 30<3E>
3838:34 C5 AE 90 0B D0 2E 88<5E>
3840:B1 8A C5 AD 90 02 D0 25<68>
3848:AD 1F D0 C9 07 F0 06 A5<16>
3850:14 29 0F D0 F3 20 07 F2<08>
```

Listing »Turbo-Basic XL« (Fortsetzung)

```
3858:A5 11 F0 0F A0 02 B1 8A<3E>
3860:18 65 8A 85 8A 90 CC E6<CD>
3868:8B B0 C8 C6 11 A5 B5 F0<E5>
3870:07 20 F8 C4 A9 00 85 B5<AC>
3878:8D FE 02 20 D3 F7 4C 2B<1E>
3880:F8 86 AA 85 96 84 95 A4<69>
3888:AA A5 AF F0 10 C6 AF B1<B6>
3890:95 30 03 C8 D0 F9 C8 20<C7>
3898:D7 F1 4C C1 F1 18 98 65<41>
38A0:95 85 95 90 02 E6 96 60<FA>
38A8:A0 FF 84 AF E6 AF A4 AF<83>
38B0:B1 95 48 C9 9B F0 04 29<59>
38B8:7F F0 03 20 96 C2 68 10<D8>
38C0:EB 60 20 02 F2 20 E2 F1<86>
38C8:A9 20 4C 96 C2 A0 00 B1<B5>
38D0:8A AA C8 B1 8A 20 52 F9<1E>
38D8:AE 91 F3 F0 11 A6 B9 A0<D5>
38E0:04 B1 8A 20 9A F3 F0 04<66>
38E8:C9 40 D0 02 CA CA 86 D4<FD>
38F0:20 02 F2 C6 D4 10 F9 A0<FC>
38F8:02 B1 8A 85 9F C8 B1 8A<22>
3900:85 A7 C8 84 A8 20 49 F2<E1>
3908:A4 A7 C4 9F 90 F0 60 20<BD>
3910:3A F3 C9 36 F0 32 C9 54<06>
3918:F0 17 20 46 F3 20 3A F3<20>
3920:C9 37 F0 04 C9 02 B0 20<10>
3928:20 38 F3 20 96 C2 4C 62<CF>
3930:F2 A0 1E AD 91 F3 D0 02<0C>
3938:A0 02 84 D4 A9 2D 20 96<01>
3940:C2 C6 D4 D0 F7 4C 94 C2<6E>
3948:20 38 F3 D0 04 20 38 F3<38>
3950:2C 10 0F 49 80 20 B1 F3<1C>
3958:C9 A8 D0 EC 20 38 F3 4C<E5>
3960:82 F2 C9 0F F0 2E B0 55<08>
3968:48 20 60 E4 C6 A8 68 C9<7D>
3970:0D D0 12 A9 24 20 96 C2<B8>
3978:20 25 33 20 05 DA 09 80<90>
3980:99 7F 05 D0 03 20 BE 2B<BB>
3988:20 5C F9 4C 82 F2 20 E2<7B>
3990:F1 4C 82 F2 20 38 F3 85<8A>
3998:AF A9 22 20 96 C2 A5 AF<08>
39A0:F0 13 20 38 F3 C9 22 D0<25>
39A8:05 20 96 C2 A9 22 20 96<37>
39B0:C2 C6 AF D0 ED A9 22 20<F5>
39B8:96 C2 4C 82 F2 38 E9 10<D9>
39C0:85 AF A2 00 A9 ED A0 71<9E>
39C8:20 BB F1 20 3A F3 C9 5A<E2>
39D0:F0 23 C9 5D F0 1F C9 61<B0>
39D8:F0 1B C9 62 F0 17 C9 6A<25>
39E0:F0 13 C9 28 F0 12 C9 3D<41>
39E8:B0 A4 A0 00 B1 95 29 7F<BC>
39F0:20 57 EA B0 99 20 02 F2<83>
39F8:20 FF F1 4C 82 F2 E6 A8<64>
3A00:A4 A8 C4 A7 B0 03 B1 8A<CB>
3A08:60 68 68 60 48 20 59 F3<7F>
3A10:68 85 AF A2 01 A9 EB A0<EA>
3A18:5A 20 BB F1 4C FF F1 C9<11>
3A20:08 F0 1F C9 3C F0 1B C9<A5>
3A28:3E F0 17 C9 45 F0 13 C9<F8>
3A30:4F F0 0F C9 07 F0 12 20<F2>
3A38:9A F3 D0 0C C6 B9 C6 B9<CF>
3A40:10 06 E6 B9 E6 B9 30 F4<EC>
3A48:60 A5 A8 48 20 66 35 68<F4>
3A50:85 A8 E0 1B D0 EC 60 01<CE>
3A58:B1 8A 49 26 8D 91 F3 60<07>
3A60:C9 09 F0 12 C9 3D F0 0E<C3>
3A68:C9 3F F0 0A C9 46 F0 06<F4>
3A70:C9 51 F0 02 C9 41 60 85<4C>
3A78:AF A2 00 A5 83 A4 82 20<FC>
3A80:BB F1 4C E2 F1 84 B3 B1<E6>
3A88:8A D0 03 C8 B1 8A 49 80<95>
3A90:85 C7 20 00 E4 A6 90 86<19>
3A98:C4 A6 91 86 C5 20 D3 F7<85>
3AA0:B0 0C D0 0A A0 0C B1 90<30>
3AA8:C5 C7 D0 F1 F0 08 A5 C4<66>
3AB0:85 90 A5 C5 85 91 A9 0D<9A>
3AB8:20 54 F6 20 6B E5 A0 00<F1>
3AC0:20 80 F6 20 BB 2F A6 A8<AB>
3AC8:E8 E4 A7 B0 03 20 6B E5<6B>
3AD0:A0 06 20 80 F6 AD 07 F7<FB>
3AD8:D0 2C A5 C7 A0 0C 91 C4<5A>
3AE0:A9 00 F0 05 4A A4 A8 84<A7>
3AE8:B3 48 A9 04 20 54 F6 68<A3>
3AF0:A0 00 91 C4 A5 8A C8 91<F6>
3AF8:C4 A5 8B C8 91 C4 A6 B3<17>
3B00:CA 8A C8 91 C4 60 A5 90<80>
3B08:48 A5 91 48 A5 C4 85 90<C9>
3B10:A5 C5 85 91 A5 D4 48 A5<EE>
3B18:C7 20 B3 E4 68 20 E5 F4<98>
3B20:90 08 68 85 91 68 85 90<F3>
3B28:B0 B0 68 68 A9 08 A2 09<18>
3B30:20 08 F7 B1 8A D0 03 C8<04>
3B38:B1 8A 49 80 45 C7 F0 C5<C5>
3B40:4C 5A F7 A9 0D 4C DC F8<36>
3B48:A5 BE 85 8A A5 BF 85 8B<98>
3B50:60 B1 8A D0 03 C8 B1 8A<94>
3B58:49 80 85 C7 20 D3 F7 B0<3A>
3B60:E2 D0 E0 A0 0C B1 90 C5<F9>
3B68:C7 D0 F1 A0 06 B1 90 48<F8>
3B70:85 E0 C8 B1 90 85 E1 C8<61>
3B78:B1 90 85 E2 C8 B1 90 85<E3>
3B80:E3 C8 B1 90 85 E4 C8 B1<75>
3B88:90 85 E5 A5 C7 20 B3 E4<1F>
3B90:20 9D 2C B0 13 20 C0 E5<6B>
3B98:68 20 E5 F4 90 40 A9 11<05>
3BA0:20 24 F5 A9 08 4C 2B F8<16>
3BA8:4C 2E E5 85 EC A0 00 B1<47>
3BB0:90 C5 D4 D0 2A C8 B1 90<74>
3BB8:C5 D5 D0 1C C8 B1 90 C5<80>
3BC0:D6 D0 15 C8 B1 90 C5 D7<5C>
3BC8:D0 0E C8 B1 90 C5 D8 D0<CC>
3BD0:07 C8 B1 90 C5 D9 F0 06<8A>
3BD8:6A 45 EC 45 D4 0A 60 05<B2>
3BE0:D4 45 EC 10 F9 6A 49 80<86>
3BE8:0A 60 18 65 90 85 90 85<FB>
3BF0:0E 90 04 E6 91 E6 0F 60<D2>
3BF8:C8 C4 A7 B0 03 20 DE C2<49>
3C00:A5 88 85 8A A5 89 85 8B<11>
3C08:A9 00 8D 07 F7 8D A3 F5<7B>
3C10:A0 03 84 A7 88 B1 8A 85<0F>
3C18:9F 88 B1 8A 30 15 20 E7<0E>
3C20:F6 20 AF F6 20 9E F6 A9<DD>
3C28:00 85 B7 85 B8 85 B6 60<A4>
3C30:20 C4 F5 4C 6B E6 20 00<3A>
3C38:E4 A6 AA BD 60 21 F0 0A<86>
3C40:A6 A8 E8 E4 A7 B0 19 4C<6B>
3C48:9D FF A4 A8 88 B1 8A C9<AA>
3C50:1B F0 09 A9 07 A2 41 A0<EF>
3C58:40 4C 0A F7 A5 9F 85 A7<C6>
3C60:60 B1 8A 49 26 8D A3 F5<5A>
3C68:60 00 A9 80 85 11 AC A3<12>
3C70:F5 F0 03 4C DC F8 20 C4<37>
3C78:F5 20 94 C2 A9 D8 85 95<4E>
3C80:A9 F5 85 96 20 E2 F1 4C<4A>
3C88:2E F9 A0 01 B1 8A 30 07<74>
3C90:85 BB 88 B1 8A 85 BA A9<32>
3C98:00 85 B4 85 B5 60 53 54<58>
3CA0:4F 50 50 45 44 A0 A0 01<C8>
3CA8:B1 8A 10 EB A5 BA 85 A0<C3>
3CB0:A5 BB 85 A1 20 62 C9 A0<20>
3CB8:02 B1 8A 85 9F 68 68 4C<84>
3CC0:D7 FF 20 3A CA 85 BC 84<6A>
3CC8:BD 60 84 B3 20 6B E5 20<E6>
3CD0:25 33 B0 1C A5 D5 D0 18<B3>
3CD8:A5 D4 F0 14 85 B9 A4 A8<CD>
3CE0:88 B1 8A 48 C6 B9 F0 09<1C>
3CE8:20 68 35 E0 12 F0 F5 68<A1>
3CF0:60 68 C9 62 F0 19 C9 6A<40>
3CF8:F0 1D 48 20 FB E4 68 C9<BB>
3D00:17 F0 05 A9 1E 20 23 F4<68>
3D08:A5 D4 A4 D5 4C A0 FF A9<B4>
3D10:51 20 23 F4 4C 85 F8 4C<8B>
3D18:AF F8 85 A4 18 A5 90 85<50>
3D20:C4 65 A4 A8 A5 91 85 C5<3C>
3D28:69 00 CD E6 02 90 09 D0<8F>
3D30:10 CC E5 02 90 02 D0 09<97>
3D38:85 91 85 0F 84 90 84 0E<92>
3D40:60 A9 02 4C DC F8 A5 D4<F6>
3D48:91 C4 C8 A5 D5 91 C4 C8<C8>
3D50:A5 D6 91 C4 C8 A5 D7 91<F8>
3D58:C4 C8 A5 D8 91 C4 C8 A5<A9>
3D60:D9 91 C4 60 A5 8C 85 BE<1C>
3D68:85 90 85 0E A5 8D 85 8F<92>
3D70:85 91 85 0F 60 A6 86 86<FF>
3D78:F5 A4 87 84 F6 A6 F5 E4<83>
3D80:88 A5 F6 E5 89 B0 23 A0<E0>
3D88:00 B1 F5 29 FC 91 F5 A0<4B>
3D90:02 A2 06 A9 00 91 F5 C8<03>
3D98:CA D0 FA A5 F5 18 69 08<D7>
3DA0:85 F5 A5 F6 69 00 85 F6<50>
3DA8:D0 D3 4C C5 C9 A0 00 84<FD>
3DB0:BA 84 BB 84 B9 84 FB 84<29>
3DB8:B6 84 B7 84 B8 88 84 BD<ED>
3DC0:84 11 4C 34 C5 B1 8A 49<94>
3DC8:26 8D 07 F7 60 00 A0 FF<4C>
3DD0:8E 30 F7 8D 34 F7 8C 38<93>
3DD8:F7 A9 00 85 D7 A5 8A 85<69>
3DE0:BE A5 8B 85 BF A4 A7 C4<C3>
3DE8:9F B0 3D B1 8A 85 A7 C8<B1>
3DF0:B1 8A C8 84 A8 C9 00 F0<07>
3DF8:0D C9 00 F0 12 C9 00 D0<AD>
3E00:E4 A5 D7 D0 E0 60 A5 D7<20>
3E08:F0 FB C6 D7 4C 1F F7 C9<77>
3E10:07 D0 09 A4 A7 88 B1 8A<B2>
3E18:C9 1B F0 C9 E6 D7 D0 C5<99>
3E20:20 82 F4 A9 16 4C DC F8<A7>
3E28:A0 01 B1 8A 30 F2 18 A5<02>
3E30:9F 65 8A 85 8A 90 02 E6<93>
3E38:8B A0 01 B1 8A 30 E1 C8<34>
3E40:B1 8A 85 9F C8 84 A7 4C<1D>
3E48:1F F7 20 1E F4 20 00 E4<F8>
3E50:A6 AA BD 60 21 D0 27 20<58>
3E58:D3 F7 A9 3E A2 3F 4C 08<9F>
3E60:F7 20 00 E4 20 D3 F7 B0<98>
3E68:28 C9 3C D0 B6 A6 AA BC<15>
3E70:60 21 D0 0A A9 04 20 24<59>
3E78:F5 A9 3C 4C 2B F8 60 20<DF>
3E80:D3 F7 B0 10 C9 3E D0 9B<B5>
3E88:20 2B F8 A4 B2 88 84 A7<7D>
3E90:60 A9 17 2C A9 18 4C DC<24>
3E98:F8 A5 8F C5 91 90 06 A5<0F>
3EA0:8E C5 90 B0 37 38 A5 90<F3>
3EA8:E9 04 85 90 85 0E B0 04<EA>
3EB0:C6 91 C6 0F A0 03 B1 90<19>
3EB8:85 B2 88 B1 90 85 A1 88<ED>
3EC0:B1 90 85 A0 88 B1 90 D0<3B>
3EC8:11 A8 38 A5 90 E9 0D 85<0F>
3ED0:90 85 0E B0 04 C6 91 C6<50>
3ED8:0F 98 18 60 38 60 20 D3<21>
3EE0:F7 B0 2D C9 0C F0 0A C9<F3>
3EE8:1E F0 06 C9 50 D0 EF F0<5C>
3EF0:1F A4 B2 D1 A0 D0 16 A5<5D>
3EF8:A1 F0 12 85 8B A5 A0 85<10>
3F00:8A 88 B1 8A 85 A7 A0 02<38>
3F08:B1 8A 85 9F 60 A9 0F 2C<5C>
3F10:A9 10 2C A9 1A 4C DC F8<21>
3F18:20 D3 F7 B0 F6 D0 02 A9<D1>
3F20:08 AA E8 4C 08 F7 A9 1E<5A>
3F28:D0 C7 20 D3 F7 B0 14 C9<DB>
3F30:50 F0 BE C9 51 F0 EF C9<00>
3F38:1E F0 08 C9 0C D0 EB 38<0C>
3F40:E9 A4 2C A9 1C 4C DC F8<DF>
3F48:20 1E F4 A2 C1 A4 A8 B1<8F>
3F50:8A D0 03 C8 B1 8A 49 80<22>
3F58:20 ED E5 8A D1 9D D0 DF<EF>
3F60:A0 03 B1 9D 85 8B 88 B1<6D>
3F68:9D 85 8A B1 8A 85 9F C8<FE>
3F70:B1 8A 85 A7 60 A2 C2 D0<DE>
3F78:D4 A9 1B 2C A9 19 4C DC<47>
3F80:F8 20 D3 F7 B0 F6 C9 45<DA>
3F88:D0 0A A9 04 20 24 F5 A9<6A>
3F90:45 4C 2B F8 4C 5D F7 A9<D3>
3F98:15 2C A9 13 2C A9 12 2C<44>
3FA0:A9 11 85 B9 A9 00 D8 8D<D5>
3FA8:FE 02 20 C4 F5 A4 BD 30<C7>
3FB0:11 A5 BC A2 80 86 BD A6<AF>
3FB8:B9 86 C3 A2 00 86 B9 4C<7A>
3FC0:A0 FF A5 B9 C9 80 D0 03<EA>
3FC8:4C B0 F5 20 94 C2 A9 37<2E>
3FD0:20 46 F3 A6 B9 A9 00 20<39>
3FD8:52 F9 A5 B9 C9 1F 90 02<D3>
3FE0:E9 62 85 AF C9 4C B0 0C<EB>
3FE8:A2 00 A9 F9 A0 67 20 BB<71>
3FF0:F1 20 FC F1 A0 01 B1 8A<6B>
3FF8:30 15 A9 67 A0 F9 20 60<5F>
4000:F9 A0 01 B1 8A 85 D5 88<2C>
4008:B1 8A 85 D4 20 56 F9 20<1B>
4010:94 C2 20 66 FF 4C 7B E6<AB>
4018:85 D5 86 D4 20 DC 2B 20<75>
4020:BE 2B A5 F3 A4 F4 85 95<F9>
4028:84 96 4C E2 F1 20 41 54<D4>
4030:20 4C 49 4E 45 A0 3F 42<3A>
4038:4C 4F 41 C4 4D 45 CD 56<7E>
4040:41 4C 55 C5 3E 23 56 41<D5>
4048:52 D3 24 4C 45 CE 3F 44<0B>
4050:41 54 C1 3E 33 32 37 36<55>
4058:B7 49 4E 50 55 D4 44 49<E0>
4060:CD 53 54 41 43 CB 4F 56<D6>
4068:45 52 46 4C 4F D7 3F 4C<AD>
4070:49 4E C5 3F 46 4F D2 54<98>
4078:4F 4F 20 4C 4F 4E C7 3F<3A>
4080:44 45 CC 3F 47 4F 53 55<3F>
4088:C2 47 41 52 42 41 47 C5<D1>
4090:3F 43 48 D2 4D 45 CD A3<1A>
4098:3F 4C 4F 41 C4 3F 4E 45<8B>
40A0:53 D4 3F 57 48 49 4C C5<FF>
40A8:3F 52 45 50 45 41 D4 3F<FF>
40B0:44 CF 3F 45 58 49 D4 58<1B>
40B8:50 52 4F C3 3F 45 58 45<0F>
40C0:C3 3F 50 52 4F C3 3F A3<37>
40C8:49 53 20 4F 50 45 CE 3F<EE>
40D0:44 45 D6 57 52 20 4F 4E<C3>
40D8:4C D9 43 4D C4 4E 4F 54<2B>
40E0:20 4F 50 45 CE A3 52 44<37>
40E8:20 4F 4E 4C D9 45 4F C6<DB>
40F0:54 52 55 4E C3 54 49 4D<83>
40F8:45 4F 55 D4 4E 41 CB 21<7C>
4100:46 52 41 4D C5 43 55 52<46>
4108:53 4F D2 21 4F 56 45 52<C2>
4110:52 55 CE 43 48 4B 53 55<21>
4118:CD 44 4F 4E C5 4D 4F 44<12>
4120:C5 4E 4F 54 20 49 4D 50<AE>
```




```
4128:CC 52 41 CD A0 A0 A0 A0<7C>
4130:A0 A0 A0 A0 A0 A0 A0 A0<1D>
4138:44 3F BA 3E 23 46 49 4C<CE>
4140:45 D3 44 53 4B 20 46 55<6C>
4148:4C CC 46 41 54 41 4C 20<0C>
4150:49 2F CF 46 49 4C 45 A3<F6>
4158:4E 41 4D C5 50 4F 49 4E<5E>
4160:D4 4C 4F 43 4B 45 C4 44<10>
4168:43 4D C4 3E 44 49 D2 3F<E5>
4170:46 49 4C C5 50 4F 49 4E<64>
4178:D4 3F 41 50 50 4E C4 42<57>
4180:41 44 20 53 45 43 54 4F<17>
4188:52 D3 A5 14 A4 13 A6 12<A4>
4190:C5 14 D0 F6 48 86 D5 84<DD>
4198:D4 20 DC 2B A2 05 BD 68<44>
41A0:FB 95 E0 CA 10 F8 20 9E<AA>
41A8:26 20 9C 2E 68 85 D4 A9<21>
41B0:00 85 D5 20 DC 2B 20 9D<58>
41B8:2C A9 00 85 D2 85 D3 4C<99>
41C0:3A E5 20 C4 FA C6 AA A2<B9>
41C8:05 BD 6E FB 95 E0 CA 10<69>
41D0:F8 20 00 28 A0 00 A5 D4<FB>
41D8:C9 40 D0 02 84 D5 84 F2<1B>
41E0:A9 24 20 30 FB A9 60 20<01>
41E8:30 FB A9 60 20 30 FB A0<53>
41F0:06 A9 80 4C 6F DA 48 20<D7>
41F8:B8 2D 20 9C 2E 20 BB 2F<FD>
4200:68 85 D5 20 9E 26 A9 00<4D>
4208:A4 D4 C0 40 D0 05 A4 D5<80>
4210:85 D5 98 AA 4A 4A 4A 4A<92>
4218:20 58 FB 8A 29 0F 09 30<4E>
4220:C9 3A 90 02 69 06 A4 F2<46>
4228:99 80 05 E6 F2 60 41 02<F1>
4230:56 00 00 00 43 04 32 00<66>
4238:00 00 20 00 E4 20 73 C5<F0>
4240:A0 00 84 F2 84 E1 84 E2<B0>
4248:20 C4 FB C9 18 B0 35 85<28>
4250:E0 20 13 FC 20 C4 FB C9<BD>
4258:3C B0 29 20 DE FB 20 13<3C>
4260:FC 20 C4 FB C9 3C B0 1C<C7>
4268:20 DE FB 20 F2 FB 20 EC<7D>
4270:FB A5 E0 A4 E1 A6 E2 85<BC>
4278:14 84 13 86 12 C5 14 D0<FE>
4280:F6 4C 93 C5 20 93 C5 4C<13>
4288:D7 F8 20 D2 2B E6 F2 B0<95>
4290:F3 0A 85 E3 0A 0A 65 E3<03>
4298:85 E3 20 D2 2B E6 F2 B0<3D>
42A0:E3 65 E3 60 18 65 E0 85<DC>
42A8:E0 90 06 E6 E1 D0 02 E6<5F>
42B0:E2 60 06 E0 26 E1 26 E2<DE>
42B8:A4 E2 A5 E0 A6 E1 06 E0<DD>
42C0:26 E1 26 E2 06 E0 26 E1<4F>
42C8:26 E2 65 E0 85 E0 8A 65<88>
42D0:E1 85 E1 98 65 E2 85 E2<6B>
42D8:60 20 EC FB A4 E2 A5 E0<17>
42E0:A6 E1 06 E0 26 E1 26 E2<21>
42E8:65 E0 85 E0 8A 65 E1 85<4D>
42F0:E1 98 65 E2 06 E0 26 E1<72>
42F8:2A 85 E2 60 60 20 0C E5<21>
4300:85 99 D0 F8 20 00 E4 A6<99>
4308:AA BD 00 21 30 03 20 34<26>
```

```
4310:DA 20 DC E4 A5 D6 05 D7<A2>
4318:F0 E2 A0 00 84 DB 84 DC<77>
4320:B1 D4 10 02 C6 DB 20 D0<40>
4328:FC 0A 0A 85 A2 A9 00 2A<AA>
4330:06 A2 2A 6D F4 02 85 A3<91>
4338:20 1A FD B0 BF A0 08 84<08>
4340:DD A4 DC B1 A2 45 DB 85<01>
4348:DA A6 ED A4 E3 06 DA B1<A2>
4350:DE 3D C0 05 90 03 1D C8<E6>
4358:05 91 DE C6 DD F0 07 20<82>
4360:0B FF C4 E1 90 E7 20 E0<F3>
4368:FE E6 DC A5 DC C9 08 B0<03>
4370:06 65 99 C5 E0 90 C6 A5<E4>
4378:9B 69 07 85 9B 90 02 E6<E0>
4380:9C E6 D4 D0 02 E6 D5 A5<3B>
4388:D6 D0 02 C6 D7 C6 D6 4C<D3>
4390:4E FC 40 20 60 00 A8 0A<7A>
4398:0A 2A 2A 29 03 AA 98 5D<D2>
43A0:CC FC 60 18 18 0C 18 30<E9>
43A8:30 60 60 C0 C0 C0 C0 18<DE>
43B0:0C C0 C0 28 14 14 0A 0A<B5>
43B8:14 14 28 28 28 28 28 28<2F>
43C0:28 14 28 00 00 00 02 03<4B>
43C8:02 03 02 03 01 01 01 00<8E>
43D0:00 03 02 00 01 03 07 00<79>
43D8:F0 FC FE 04 02 01 38 60<39>
43E0:A5 57 29 0F AA A9 00 85<D9>
43E8:DF A5 99 0A 26 DF 0A 26<C5>
43F0:DF 65 99 90 02 E6 DF 0A<CA>
43F8:85 DE 26 DF BD DD FC 85<01>
4400:E0 C5 99 F0 D9 90 D7 BC<CB>
4408:FD FC 84 E2 B9 11 FD 8D<4D>
4410:C0 05 BD ED FC 85 E1 4A<A2>
4418:4A 4A 4A AA F0 07 06 DE<B1>
4420:26 DF CA D0 F9 18 A5 DE<F9>
4428:65 58 85 DE A5 DF 65 59<AC>
4430:85 DF A5 9C 85 E4 A5 9B<21>
4438:85 E3 A4 E2 84 ED F0 0C<5A>
4440:39 0D FD 85 ED 46 E4 66<42>
4448:E3 88 D0 F9 A5 E4 D0 8E<F0>
4450:A5 E3 C5 E1 B0 88 A6 E2<79>
4458:D0 0A A5 C8 20 D0 FC 8D<41>
4460:C8 05 18 60 BC 0D FD 84<A7>
4468:E4 BD 14 FD 85 EE A5 C8<53>
4470:0D C0 05 4D C0 05 99 C8<F0>
4478:05 A6 EE 0A CA D0 FC 88<98>
4480:10 F4 A4 E4 AD C0 05 99<D7>
4488:C0 05 A6 EE 38 2A CA D0<13>
4490:FB 88 10 F3 18 60 20 0C<77>
4498:E5 85 99 D0 F8 20 1A FD<71>
44A0:B0 F3 A5 90 85 A2 A5 91<27>
44A8:85 A3 AD E5 02 E9 06 85<D4>
44B0:E7 AD E6 02 E9 00 85 E8<EC>
44B8:18 A5 A2 69 03 85 A2 90<73>
44C0:02 E6 A3 C5 E7 A5 A3 E5<77>
44C8:E8 90 03 4C 7B F6 A6 ED<71>
44D0:A4 E3 20 26 FF F0 03 4C<53>
44D8:BA FE 20 1B FF 20 14 FF<F1>
44E0:98 30 0B 20 26 FF D0 06<36>
44E8:20 1B FF 4C 17 FE 20 0B<67>
44F0:FF 98 A0 00 91 A2 8A 0A<03>
```

```
44F8:0A 0A C8 91 A2 A4 E3 A6<29>
4500:ED 20 0B FF C4 E1 B0 0B<6A>
4508:20 26 FF D0 06 20 1B FF<EE>
4510:4C 3B FE 20 14 FF 98 A0<AD>
4518:02 91 A2 8A 88 11 A2 91<09>
4520:A2 A4 99 C8 C4 E0 B0 23<B6>
4528:20 DE FE 20 FA FE A0 01<46>
4530:B1 A2 29 07 C5 ED C8 B1<53>
4538:A2 E5 E3 90 0B A0 00 B1<A9>
4540:A2 09 80 91 A2 4C F2 FD<F1>
4548:20 EC FE A4 99 88 C4 E0<26>
4550:B0 20 20 EC FE 20 FA FE<DF>
4558:A0 01 B1 A2 29 07 C5 ED<FF>
4560:C8 B1 A2 E5 E3 90 08 A0<C7>
4568:00 B1 A2 29 7F 10 D4 20<E3>
4570:DE FE A0 01 B1 A2 29 07<F2>
4578:AA C8 B1 A2 A8 20 0B FF<74>
4580:20 0B FF 86 ED 84 E3 38<8C>
4588:A5 A2 E9 03 85 A2 B0 02<AE>
4590:C6 A3 C5 90 D0 06 A5 A3<5C>
4598:C5 91 F0 15 A0 00 B1 A2<40>
45A0:10 B6 30 8A E6 99 18 A5<98>
45A8:DE 65 E1 85 DE 90 02 E6<06>
45B0:DF 60 C6 99 38 A5 DE E5<DA>
45B8:E1 85 DE B0 02 C6 DF 60<CF>
45C0:A0 00 B1 A2 29 7F 85 E3<89>
45C8:C8 B1 A2 4A 4A 4A 85 ED<CE>
45D0:60 E4 E4 E8 90 03 A2 00<EB>
45D8:C8 60 CA 10 03 A6 E4 88<9B>
45E0:60 B1 DE 3D C0 05 1D C8<E8>
45E8:05 91 DE 60 B1 DE 1D C0<D4>
45F0:05 5D C0 05 F0 04 AD C8<0D>
45F8:05 60 AD C8 05 F0 03 A9<D6>
4600:00 60 A9 01 60 20 45 DF<D6>
4608:AD FE 02 48 8E FE 02 20<38>
4610:9C C4 A0 7D 20 9C C2 68<EE>
4618:8D FE 02 4C A9 C2 B1 8A<29>
4620:C9 26 F0 08 A9 4C A2 76<50>
4628:A0 FF D0 06 A9 A0 A2 02<CD>
4630:A0 B1 8D AF FF 8E B0 FF<06>
4638:8C B1 FF 60 A9 5B 20 96<65>
4640:C2 A0 00 B1 8A AA C8 B1<E0>
4648:8A 20 52 F9 A9 5D 20 96<0C>
4650:C2 A0 02 B1 8A 4C B3 FF<DC>
4658:20 82 F4 A9 0C 4C DC F8<AF>
4660:20 1E F4 20 FB E4 85 A0<D7>
4668:84 A1 20 62 C9 B0 E9 68<2F>
4670:68 A5 11 F0 1E A0 02 B1<1F>
4678:8A 85 9F C8 C4 9F B0 1D<34>
4680:B1 8A 85 A7 C8 B1 8A C8<42>
4688:84 A8 20 D0 FF A4 A7 A5<BE>
4690:11 D0 E9 4C A4 F5 0A 8D<14>
4698:D5 FF 6C 00 22 A0 01 B1<22>
46A0:8A 30 12 18 A5 9F 65 8A<2D>
46A8:85 8A 90 02 E6 8B B1 8A<B0>
46B0:10 C3 4C 6A F5 4C 78 E6<05>
Laenge 18104 Bytes
```

Listing »Turbo-Basic XL« (Schluß)







Schnell, schneller, Turbo-Basic XL aus dem Compiler

Passend zum Interpreter für Turbo-Basic XL gibt es jetzt noch das Turbo-Basic XL als Compiler zum Abtippen. Er macht langsame Basic- und Turbo-Basic-XL-Programme superschnell.

Immer dann, wenn es bei Basic-Programmen auf die Geschwindigkeit ankommt, ist eine optimale Programmierung erforderlich. Man sollte unnötige Sprünge vermeiden und weniger benutzte Programmteile ans Ende verbannen. Dies ist aber nur als goldene Regel gedacht. Liegt aber ein optimales Programm vor, das noch schneller werden soll, dann hilft nur ein Basic-Compiler weiter. Dieser wandelt normale Basic-Programme in Maschinensprache um, bevor sie gestartet werden. Die Programme sind dann natürlich entsprechend schneller. Auf Seite 32 finden Sie übrigens einen Vergleich zwischen den gängigsten Compilern und dem Turbo-Basic-Compiler. Der hier abgedruckte Compiler hat bei der Gegenüberstellung die besten Zeiten erzielt. Wenn man dann noch in Betracht zieht, daß es den Turbo-Basic-Compiler fast zum Nulltarif gibt, dann ist er ein absolutes Muß.

Ein Sprinter stellt sich vor: Compiler kontra Interpreter

Die heutigen Mikroprozessoren sind nicht in der Lage, ein Basic-Programm direkt auszuführen. Sie kennen nur ihre interne Sprache, also die Maschinensprache. Aus diesem Grund ist der sogenannte Interpreter »zwischengeschaltet«. Er wandelt jede Basic-Programmzeile wieder in den Maschinencode um. So ist es einsichtig, daß Basic-Programme relativ langsam sind. Ein einfacher Basic-Befehl, wie beispielsweise »PRINT 3*4« kann nämlich in der Übersetzung schon aus mehreren tausend Maschinenbefehlen bestehen.

Nun gibt es zwei Wege, ein Programm in einer Hochsprache, dazu zählt auch Basic, auszuführen. Entweder speichert man jeden Befehl, so wie er eingegeben wird. Nach dem Starten mit RUN führt ein entsprechendes Programm, der Interpreter also, für jeden einzelnen Befehl eine entsprechende Maschinensprache-Routine aus. Oder aber das gesamte Programm wird erst in Maschinensprache übersetzt, also compiliert, bevor es ausgeführt wird. Beide Verfahren haben Vor- und Nachteile.

Für den Interpreter spricht die einfache Bedienung. Es lassen sich mit ihm problemlos Programmänderungen durchführen, die dann auch schnell zu testen sind.

Mit dem Compiler zu arbeiten ist etwas umständlicher und die Fehlersuche aufwendiger. Hat man nämlich in seinem Basic-Programm einen nicht compilierbaren Befehl verwendet, dann muß man zwangsläufig den Interpreter wieder laden oder den Computer ausschalten und erneut booten. Diesen Nachteil gleichen aber compilierte Programme gegenüber normalen Basic-Programmen durch ihre höhere Geschwindigkeit wieder aus.

Ein Interpreter benötigt weniger Speicherplatz als ein Compiler. Ebenso ist fast für jeden Compiler ein schneller Massenspeicher, also ein Diskettenlaufwerk, ein Muß. Es gibt sogar Compiler, die zwei Diskettenlaufwerke voraussetzen: eines für den Compiler und ein weiteres für das zu compilierende Programm. Der Turbo-Basic-Compiler kommt allerdings mit nur einem Laufwerk aus.

Der Atari-Basic-Interpreter gehört einer besonders langsamen Spezies an. Als nämlich im Jahre 1979 der Interpreter entwickelt wurde, mußte er unbedingt in einem 8 KByte großen ROM Platz finden. Speicherplatz war zu der Zeit noch sehr teuer. Weiterhin rechnet das Atari-Basic mit BCD-Zahlen (Binary Coded Decimals), um Rundungsfehler zu vermeiden. Dies ist prinzipiell zeitaufwendiger als die normale binäre Arithmetik.

Turbo-Basic XL nimmt 18 KByte in Anspruch. Es enthält spezielle BCD-Routinen. Solche Algorithmen sind zwar recht umfangreich, dafür aber sehr schnell.

Compiler zweigeteilt

Der Turbo-Basic-Compiler besteht aus zwei getrennten Programmen. Erstens aus dem eigentlichen Compiler, der die Übersetzung von Basic-Programmen übernimmt, und zweitens aus der Runtime-Bibliothek. Dieser Programmteil enthält die Arithmetik- sowie andere Routinen, die zum Lauf des Compilats nötig sind. So ist sichergestellt, daß sich auch lange Programme compilieren lassen. Zum Zeitpunkt des Compilierens müssen sich also nicht der Compiler und das Runtime-Paket gleichzeitig im Speicher befinden. Weiterhin spart man sich so auch Diskettenplatz.

Der Compiler ist mit seinen etwa 20000 Byte sehr umfangreich. Um nun überhaupt damit arbeiten zu können, sind noch einige Dinge vorauszuschicken. Sie benötigen unbedingt ein Diskettenlaufwerk (mit Kassettenrecorder funktionieren weder Compiler noch Turbo-Basic XL). Außerdem müssen Sie, bevor Sie mit dem Abtippen der beiden Programme beginnen, unbedingt AMPEL (Atari-Maschinen-Programm-Eingabe-Listing) eingeben. Von Basic aus ist eine Eingabe des Compilers nämlich nicht möglich.

Mit AMPEL aber lassen sich reine Maschinenprogramme sehr schnell abtippen. Außerdem reduziert dieses Eingabeprogramm die Anzahl der einzugebenden Zeichen auf etwa 45000. Entsprechende DATA-Listings oder Basic-Lader würden mehr als 80000 Zeichen umfassen, und das ohne Fehlerkontrolle.

Wenn Sie also das Eingabeprogramm »AMPEL« vorliegen haben, starten Sie mit RUN und RETURN. Wählen Sie dann einen Namen, unter dem Sie das einzugebende Programm speichern möchten. Die Gerätebezeichnung ist bereits vor-

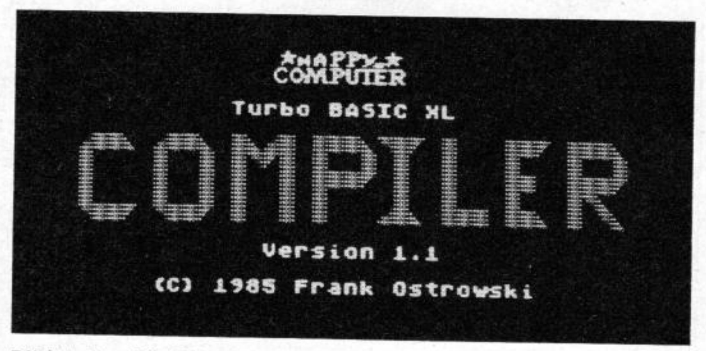


Bild 1. Das Titelbild des Turbo-Basic-Compilers

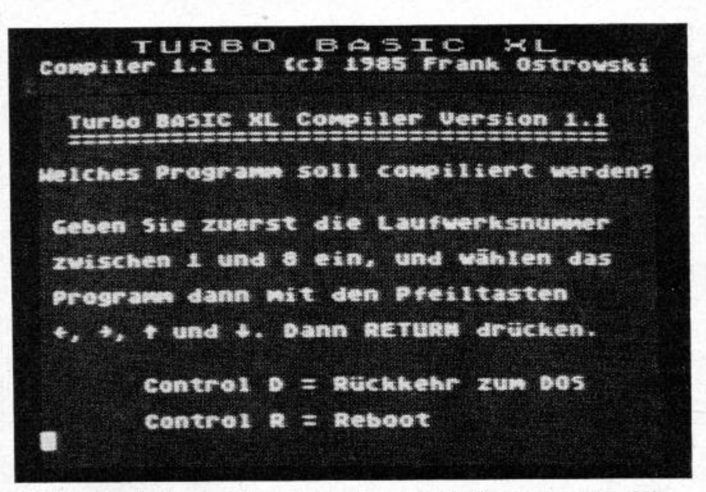


Bild 2. Nach dem Laden erscheint das Erklärungsmenü. Mit der Taste 1 können Sie sich das Inhaltsverzeichnis der Diskette in Laufwerk Nummer 1 betrachten.

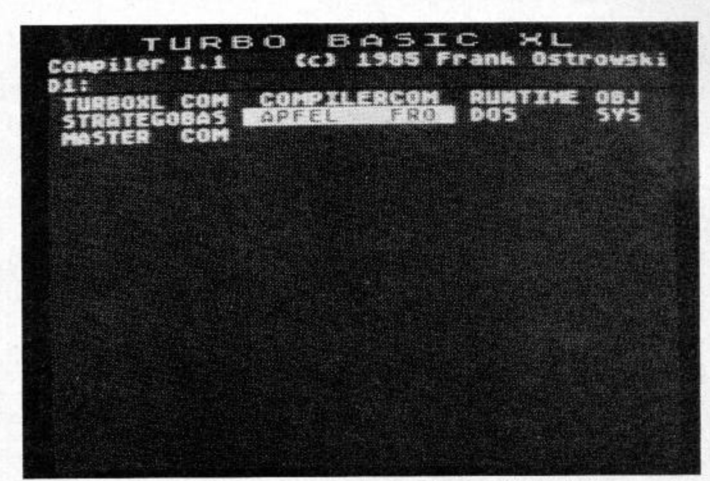


Bild 3. Das Inhaltsverzeichnis präsentiert sich in dieser Form. Mit den Pfeiltasten wählen Sie das zu compilierende Programm aus.

PROGRAMM-STECKBRIEF	
Programmname	Turbo-Basic-Compiler
Programmtyp	Programmiersprache
Programmiersprache	Maschinensprache
Programmlänge	20809 Byte
für Computer	800 XL/130XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	AMPEL
Bemerkung	Der Compiler ist in zwei Teile aufgeteilt. Einerseits in den Compiler und andererseits in das Runtime-Paket
Leserservice	Diskette (COMPILER.COM und RUNTIME.COM)

gegeben, nämlich »D1:« für Diskettenlaufwerk Nummer eins. Anschließend werden Sie nach der Anzahl der Bytes gefragt. Entnehmen Sie diese Zahl vom Ende des jeweiligen AMPEL-Listings. Sollten Sie bereits einen Teil des Programms vorliegen haben, geben Sie bitte unbedingt den zuvor gewählten Namen ein. Dann wird der bereits eingegebene Teil, den Sie mit CONTROL-S gespeichert haben, wieder in den RAM-Speicher geladen, und Sie können an der Stelle fortfahren, an der Sie aufgehört hatten.

Bedenken Sie bitte, daß Sie bei der Benennung des Programms unterschiedliche Namen verwenden müssen. Wählen Sie also für Listing 1 beispielsweise den Namen »COMPILER.OBJ« und für das zweite »RUNTIME.OBJ«. Somit sind eventuelle Verwechslungen weitestgehend ausgeschlossen.

Wenn beide Programme komplett eingegeben sind, sollten Sie sich unbedingt eine oder besser noch mehrere Sicherheitskopien anfertigen. Auch ist zu empfehlen, während der Eingabe der Programme öfter CONTROL-S zu betätigen, um das Programm im Speicher auf Diskette zu speichern. Das dauert nur wenige Sekunden, für das erneute Eingeben der Programme nach einem Stromausfall benötigen Sie jedoch ein Vielfaches der Zeit.

Verwenden Sie auch für den eigentlichen Compiler und das Runtime-Programm jeweils eine eigene Diskette. Beide Programme sind nämlich als »AUTORUN.SYS«-Files ausgelegt. Das bedeutet, daß sich beide Programme nach dem Einschalten des Computers automatisch laden und starten können. Für ein Autostart-Programm ist Voraussetzung, daß es die Bezeichnung »AUTORUN.SYS« auf Diskette hat. Dann muß sich nur noch das File DOS.SYS (entweder DOS 2.0 oder DOS 2.5) und gegebenenfalls DUP.SYS auf der Diskette befinden, und schon laden und starten sich die Programme nach dem Einschalten des Computers automatisch.

Das Umbenennen eines Dateinamens ist übrigens sehr einfach. Nehmen wir an, daß sich der Compiler unter dem Namen »COMPILER.OBJ« auf Diskette befindet. Das Programm soll nun in »AUTORUN.SYS« umbenannt werden. Legen Sie also eine Diskette mit den Files DOS.SYS und DUP.SYS ins Laufwerk ein, und schalten Sie die Stromversorgung des Computers an. Nach der READY-Meldung geben Sie DOS ein, um in das entsprechende Menü zu gelangen. Wählen Sie jetzt die Funktion E für »Rename File«, und geben Sie den alten und, mit einem Komma getrennt, den neuen Programmnamen ein. Der neue Programmname lautet also in unserem Beispiel »AUTORUN.SYS«. Jetzt haben Sie ein Autostart-Programm vorliegen. Dies funktioniert jedoch nur mit speziell für einen Autostart ausgelegten Maschinen-Programmen.

Für Ihre Mühe, das Programm eingetippt zu haben, werden Sie – falls Sie auch an genügend Sicherheitskopien gedacht haben – belohnt. Sie verfügen nämlich damit über einen Compiler, der sowohl beim Compilieren als auch bei der Programmausführung besonders schnell ist.

Bedienung des Compilers

Der Umgang mit dem Turbo-Basic-Compiler ist sehr einfach. Es ist nicht nötig, laufend mit der Bedienungsanleitung in der Hand zu arbeiten. Allerdings sollten Sie sich diese Beschreibung zumindest einmal aufmerksam durchlesen. Übrigens, haben Sie schon Ihren Compiler und das Runtime-Paket auf jeweils einer Diskette vorliegen?

Wenn Sie also ein Programm compilieren möchten, sollte es mit SAVE gespeichert sein. Mit LIST gespeicherte Programme lassen sich mit dem Turbo-Basic-Compiler nicht compilieren.

Schieben Sie also einfach die Compiler-Diskette in Laufwerk Nummer D1:, und schalten Sie anschließend den Computer ein. Dann wird automatisch das DOS und das AUTORUN.SYS-File mit dem Compiler geladen. Zunächst sollte das Titelbild (Bild 1) erscheinen. Dieser Compiler meldet sich mit einer kurzen Bedienungsanleitung (Bild 2). Jetzt können Sie die Compiler-Diskette aus dem Laufwerk nehmen, sie wird für den folgenden Vorgang nicht mehr benötigt. Legen Sie dann die Diskette mit dem zu compilierenden Programm ein, und drücken Sie die Taste 1. Sollte Ihnen mehr als ein Laufwerk zur Verfügung stehen, geben Sie die entsprechende Laufwerknummer an. Anschließend wird der Bildschirm gelöscht. Es verbleiben nur drei Zeilen am oberen

Bildschirmrand. Auf dem Bildschirm erscheint daraufhin das Inhaltsverzeichnis der Diskette. Aus Platzmangel mußte auf die Angabe der belegten Sektoren der einzelnen Files verzichtet werden. Sollte die Diskette leer sein, gelangen Sie wieder zum Titelbild. Wenn sich das zu compilierende Programm nicht auf Diskette befindet, wechseln Sie sie einfach aus und betätigen dann noch einmal die Taste 1.

Der erste Programmname, in der oberen linken Bildschirmecke, wird invers dargestellt. Bild 3 verdeutlicht dies. Die Ausgabe erfolgt dabei mit dunklen Buchstaben auf hellem Hintergrund, so als wäre der Cursor viel breiter. Diesen »Cursor« können Sie mit den Pfeiltasten, ohne CONTROL-Taste, bewegen. Wenn der richtige Filename invertiert ist, drücken Sie RETURN, und schon beginnt die Compilierung. Während dieses Vorgangs wird nur von Diskette gelesen, aber keine Zwischenfiles auf Diskette abgelegt.

Beim Turbo-Basic-Compiler handelt es sich um einen »Two Pass Compiler«. Das heißt, die Compilierung erfolgt in zwei Durchgängen. Dabei muß das Programm nur einmal von Diskette gelesen werden. War das Programm problemlos zu compilieren, erfolgen die Meldung »Keine Fehler« und eine Angabe über die Programmlänge. Die Frage nach dem Programmnamen, unter dem das compilierte Programm auf Diskette gespeichert werden soll, schließt das Ganze ab. Dabei sind die Gerätebezeichnung, also D1 für Laufwerk Nummer eins, und der Fileextender (der Text, der nach dem Punkt folgt), also ».FBC« schon vorgegeben. Wenn Sie mehr als ein Laufwerk besitzen, können Sie an dieser Stelle die entsprechende Taste für die Laufwerknummer drücken.

Geben Sie jetzt den Filenamen ein. Das erste Zeichen muß dabei ein Buchstabe sein, und die Gesamtlänge darf acht Zeichen nicht überschreiten. Bei einer fehlerhaften Eingabe können Sie mit der DELETE/BACKSPACE-Taste korrigieren. Nach Betätigung der RETURN-Taste wird das endgültige Compilat unter dem eingegebenen Namen auf der Diskette gespeichert, worauf sich auch die Runtime-Bibliothek (als AUTORUN.SYS-File) befinden sollte. Wenn Sie Ihrem compilierten Programm den Namen AUTORUN.FBC geben, wird es beim Booten automatisch geladen und gestartet.

Ist das Speichern beendet, werden Sie gefragt, ob Sie das Programm noch einmal speichern möchten. Drücken Sie »J« für ja, sonst »N« für nein. Mit »N« gelangen Sie wieder ins Titelbild, das nach dem Booten des Compilers auf dem Bildschirm zu sehen war. Jetzt können Sie wieder die Taste 1 drücken, falls Sie ein weiteres Programm compilieren möchten. Mit CONTROL-D gelangt man ins DOS, und CONTROL-R entspricht dem Aus- und Einschalten des Computers.

Zum Ausführen des compilierten Programms müssen Sie die Diskette mit der Runtime-Bibliothek einlegen und booten. Dazu können Sie die CONTROL-R-Funktion im Compiler wählen oder den Computer aus- und wieder einschalten. Wenn sich auf der Diskette ein File mit dem Namen AUTORUN.FBC befindet, wird dieses geladen und gestartet. Andernfalls erscheint die Meldung »Fehler 170 in 12345 ($AAAA)«. Die Fehlernummer 170 steht für »File not found«. Es wurde also versucht, ein nicht auf Diskette befindliches Programm zu laden. Nähere Informationen hierzu kann man auch der Basic- oder DOS-Anleitung entnehmen. Sollte also diese Fehlermeldung erscheinen, ist die Zeilennummer zufällig und der in Klammern angegebene 6502-Programmzählerstand unwichtig. Nach jeder Fehlermeldung, dem Programmende oder END, STOP und DOS-Befehlen in compilierten Programm erscheint »Programmende: Dos, Run oder Load?«. Wenn Sie jetzt die Taste D drücken, wird das DOS aufgerufen, R wirkt wie RUN vom Basic-Interpreter aus, und L lädt ein Programm von Diskette (wie »RUN "D:Filename.ext"« unter Basic). Alle anderen Tasten werden ignoriert. Nach L erscheint »"Filename D:"« auf dem Bildschirm. Sie können jetzt den Namen eines compilierten Programms eingeben. Der Extender ».FBC« wird automatisch angefügt, allerdings nicht angezeigt (sonst wäre das Runtime-File zu lang). Wie unter DOS können Sie auch ein Prefix wie D2: angeben.

Nicht compilierbare Befehle

Es gibt eine Reihe von Befehlen, die der Turbo-Basic-Compiler nicht unterstützt. Diese sind im einzelnen: LIST, ENTER, *L, DEL, RENUM, DUMP, TRACE, CONT, LOAD, SAVE, CLOAD, CSAVE, NEW und ERROR. Sämtliche Variablennamen sowie REM-Zeilen werden beim Compilieren nicht berücksichtigt.

Andererseits compiliert der Turbo-Basic-Compiler eine Reihe von Befehlen, die andere, wie beispielsweise der ABC- oder MMG-Compiler, nicht unterstützen. Dabei handelt es sich besonders um die Turbo-Basic XL-spezifischen Befehle, berechnete GOTO-Anweisungen, GOSUB, TRAP und RESTORE sowie DIM. Berechnete Sprungziele nach ON werden allerdings nicht verarbeitet.

Außerdem stellt der Turbo-Basic-Compiler noch gewisse Ansprüche an die Strukturierung von Programmen. So dürfen in einem Programm einer FOR-NEXT-Schleife auf keinen Fall mehrere NEXT-Anweisungen zugeordnet sein. Das gleiche gilt für die REPEAT-UNTIL-, WHILE-WEND- und DO-LOOP-Schleifen. Sie dürfen weiterhin nicht durch GOTO-Anweisungen zerstückelt werden. Fehlermeldungen wie »110 FOR fehlt«, »120 NEXT fehlt«, »800 UNTIL fehlt« sind auf solche ungünstigen Programm-Konstruktionen zurückzuführen.

Hier zwei Beispiele für einen sehr ungeschickten Programmierstil:

```
100 GOTO 120
110 NEXT I:GOTO 130
120 FOR I=1 TO 5:PRINT I:GOTO 110
130 REM
```

```
100 FOR I=1 TO 10
110 IF A THEN ? -I:NEXT I:GOTO 130
120 ? I:NEXT I
130 REM
```

Diese beiden Programme werden vom Interpreter anstandslos ausgeführt. Beim Compilieren ist aber die Reihenfolge der FOR- und NEXT-Befehle im Programmtext ausschlaggebend und nicht die Reihenfolge, die sich bei Verwendung von GOTO-Anweisungen ergibt. Der Turbo-Basic XL-Befehl GO # (GOTO Marke) ist in den oben gezeigten Beispielen ebenfalls nicht zulässig.

Das erste Programm ist ein typisches Beispiel für den sogenannten »Spaghetticode«. Der Programmfluß läßt sich ähnlich gut verfolgen wie die Schlingen einer Nudel auf einem Teller. Das zweite Beispiel ist schon besser programmiert. Nutzt man aber die Fähigkeiten von Turbo-Basic XL, nämlich strukturiert zu programmieren, voll aus, sähe das zweite Beispiel so aus:

```
100 FOR I=1 TO 10:IF A:? -I:ELSE:? I:ENDIF:NEXT I
```

oder, noch übersichtlicher, so:

```
100 FOR I=1 TO 10
110   IF A
120     ? -I
130   ELSE
140     ? I
150   ENDIF
160 NEXT I
```

In Atari-Basic könnte das dann so aussehen:

```
100 FOR I=1 TO 10
110 IF A THEN ? -I:GOTO 125
120 ? I
125 NEXT I
130 REM
```

Auch das Atari-Basic-Beispiel wird vom Compiler anstandslos verarbeitet und erzeugt sogar genau den gleichen Code wie die strukturierte Turbo-Basic XL-Version. Die Übersichtlichkeit leidet aber hier schon durch den GOTO-Befehl in Zeile 110. Sie sollten wirklich nur dann so programmieren, wenn Sie auf diese Weise Ihr Programm besser überblicken können. Möchte man aber sein Programm durchgehend strukturiert gestalten, dürfen keine GOTO- sowie GOSUB-Befehle verwendet werden. Wer bereits Erfahrungen mit der Programmiersprache Pascal gesammelt hat, weiß, daß es hier solche Befehle überhaupt nicht gibt. In Turbo-Basic XL lassen sich die GOTO- und GOSUB-Befehle fast immer durch IF-ELSE-ENDIF, WHILE-WEND, REPEAT-UNTIL oder DO-EXIT-LOOP ersetzen.

Weitere Fehler treten auf, wenn bei POKE-, DPOKE-, PEEK- oder auch PLOT- oder GRAPHICS-Befehlen zu große, oder zu kleine Zahlen verwendet werden; also bei allen Befehlen, die Integer-Parameter verwenden. Zum Beispiel: »POKE 100,300«, »DPOKE 70000,1«, »PLOT 100,300«, »? PEEK (-1)« oder »GRAPHICS -5« sind also nicht zulässig. Der Turbo-Basic-Compiler wandelt nämlich bei solchen Befehlen konstante Zahlen selbständig in die kompaktere Integer-Darstellung um. Dies erledigt der Basic-Interpreter erst zur Laufzeit und gibt dann die Fehlermeldung »ERROR 3« aus. Diese Umwandlung kostet den Interpreter viel Zeit und ist einer der Gründe dafür, warum compilierte Programme schneller sind. Als Fehlermeldungen gibt es noch »Integer <0 oder >65535« und »Integer >255«.

Für Atari-Computer werden neben dem Standard-Atari-Basic und Turbo-Basic XL noch einige weitere Basic-Interpreter angeboten. Dazu zählen Basic A+, Basic XL und Basic XE. Diese Interpreter verwenden andere Token als Turbo-Basic XL. Sollten Sie also ein Programm, das mit einem dieser Interpreter geschrieben wurde, compilieren, verwechselt der Turbo-Basic-Compiler zwangsläufig die Token, und somit die Befehle. Als Resultat erhalten Sie dann ein vollkommen falsches Compilat. In den meisten Fällen bekommen Sie dann die Meldung »Compiler Fehler« oder »Token #xxx ***Muell*«.

Eine weitere Fehlermeldung ist »Ausdruck zu komplex«. Sie tritt auf, wenn bei der Übersetzung einer Berechnung ein spezieller, interner Umwandlungspuffer des Compilers überläuft. Dieser Fehler ist aber – auch nach intensivem Testen des Compilers – noch nicht aufgetreten.

(Frank Ostrowski/wb)

Vorsicht, Fehler QJSA und QJSB!

Bei der Entwicklung des Turbo-Basic-Compilers wurden die Fehlermeldungen »QJSA« und »QJSB« eingebaut. Diese dürfen aber unter keinen Umständen auftreten. Sollten Sie solche Fehler erhalten, teilen Sie dies bitte umgehend der Happy-Computer-Redaktion mit. Legen Sie dann Ihr Programm auf Diskette bei. Wenn Sie den Turbo-Basic-Compiler abgetippt haben, kann auch ein unentdeckter Tippfehler die Ursache sein. Wenden Sie sich aber auf jeden Fall an:

Redaktion Happy-Computer
z. Hd. Werner Breuer
»Compiler-Fehler«
Hans-Pinsel-Str. 10a
8013 Haar

```
0000:FF FF 00 3E AF 3E 00 18<1A>
0008:18 FF 3C 3C 66 42 00 00<53>
0010:00 06 66 7E 76 66 00 00<7B>
0018:18 3C 66 7E 66 66 00 3E<D6>
0020:73 33 3E 30 30 78 00 7C<3C>
0028:37 33 3E 30 30 78 00 00<BE>
0030:60 63 36 3C 39 63 00 01<0C>
0038:01 0F 03 03 F6 E4 00 80<CC>
0040:80 F0 C0 C0 60 20 00 01<B4>
0048:03 06 06 06 03 01 00 E1<69>
0050:33 06 06 06 33 E1 00 C7<70>
0058:63 33 36 36 66 CE 00 0E<E8>
0060:9C 9C F6 66 66 67 00 7E<89>
0068:33 33 3E 30 30 78 00 67<DF>
0070:66 66 66 66 66 3C 00 FF<EA>
0078:31 31 31 31 31 7B 00 FB<64>
0080:81 81 F1 81 81 FB 00 F8<CF>
0088:8C 8C F8 B0 98 CC 55 AA<3E>
0090:55 AA 55 AA 55 AA 01 02<50>
0098:05 0A 15 2A 55 AA 00 80<94>
00A0:40 A0 50 A8 54 AA 55 AA<17>
00A8:54 A8 50 A0 40 80 55 2A<B5>
00B0:15 0A 05 02 01 00 00 30<B6>
00B8:62 31 A2 00 BD 00 E0 9D<AD>
00C0:00 3C BD 00 E1 9D 00 3D<1D>
00C8:BD 00 E3 9D 00 3F E8 D0<FA>
00D0:EB A9 10 8D C6 02 A9 3C<4F>
00D8:8D C5 02 8D F0 02 8D F4<FE>
00E0:02 A9 47 8D 44 03 A9 30<7D>
00E8:8D 45 03 A9 1C 8D 48 03<E5>
00F0:A9 01 8D 49 03 A9 0B 8D<29>
00F8:42 03 20 56 E4 8E F0 02<EA>
0100:60 7D 9B 9B 7F 7F 20 00<8C>
0108:01 02 03 04 05 06 07 9B<94>
0110:7F 7F 08 09 0A 0B 0C 0D<E4>
0118:0E 0F 10 9B 9B 20 20 20<4C>
0120:20 20 20 20 20 20 20 20<11>
0128:54 75 72 62 6F 20 42 41<CF>
0130:53 49 43 20 58 4C 9B 9B<5A>
0138:20 12 11 11 13 20 12 11<50>
0140:11 13 20 11 13 12 11 20<C7>
0148:11 11 11 13 15 11 14 20<9F>
0150:11 20 20 20 11 11 14 20<FD>
0158:11 11 11 13 9B 20 11 20<19>
0160:20 20 20 11 20 20 11 20<4B>
0168:11 15 14 11 20 11 20 20<71>
0170:11 20 11 20 20 11 20 20<88>
0178:20 11 20 20 20 11 20 20<F9>
0180:11 9B 20 11 20 20 20 20<E0>
0188:11 20 20 11 20 11 20 20<C5>
0190:11 20 11 11 11 14 20 11<80>
0198:20 20 11 20 20 20 11 11<87>
01A0:20 20 11 11 11 14 9B 20<1D>
01A8:11 20 20 20 20 11 20 20<D6>
01B0:11 20 11 20 20 11 20 11<B5>
01B8:20 20 20 20 11 20 20 11<31>
01C0:20 20 20 11 20 20 20 11<BA>
01C8:15 13 20 9B 20 15 11 11<48>
01D0:14 20 15 11 11 14 20 11<C2>
01D8:20 20 11 20 11 20 20 20<7C>
01E0:12 11 13 20 11 11 11 20<A3>
01E8:11 11 13 20 11 20 15 13<62>
01F0:9B 9B 7F 7F 56 65 72 73<A5>
01F8:69 6F 6E 20 31 2E 31 9B<DA>
0200:9B 7F 20 28 43 29 20 31<68>
0208:39 38 35 20 46 72 61 6E<82>
0210:6B 20 4F 73 74 72 6F 77<A2>
0218:73 6B 69 9B 9B E2 02 E3<CE>
0220:02 00 30 7D 20 98 21 20<B4>
0228:40 15 A9 7D A0 20 85 0C<11>
0230:84 0D A9 00 85 52 85 A2<E5>
0238:A9 FE 8D 01 D3 4C 00 C0<39>
0240:EE 01 D3 6C 0A 00 EE 01<C9>
0248:D3 4C 77 E4 44 31 3A 2A<37>
0250:2E 2A 9B 50 3A 9B 45 3A<4B>
0258:9B EE 01 D3 20 56 E4 08<98>
0260:48 AD 01 D3 29 FC 09 02<0C>
0268:8D 01 D3 68 28 60 68 AA<01>
0270:CE 01 D3 68 40 2C 0F D4<21>
0278:10 03 6C 00 02 48 8A 48<DE>
0280:A9 20 48 A9 C4 48 BA BD<62>
0288:05 01 48 D8 48 8A 48 98<62>
0290:48 EE 01 D3 8D 0F D4 6C<3F>
0298:22 02 48 A9 20 48 A9 C6<FB>
02A0:48 08 EE 01 D3 6C 16 02<13>
02A8:EE 01 D3 20 08 21 CE 01<B1>
02B0:D3 60 BD 47 03 48 BD 46<2A>
02B8:03 48 BD 4A 03 85 2A BD<8B>
02C0:4B 03 85 2B 98 A0 5C 60<E0>
02C8:EE 01 D3 06 2A 20 E2 F6<62>
02D0:CE 01 D3 60 EE 01 D3 20<95>
02D8:D2 D9 CE 01 D3 A5 D4 A4<7E>
02E0:D5 60 70 70 40 46 5D 21<71>
02E8:00 02 00 02 00 42 B8 BC<C2>
02F0:02 02 02 02 02 02 02 02<F4>
02F8:02 02 02 02 02 02 02 02<FC>
0300:02 02 02 02 41 38 21 00<0D>
0308:00 00 34 35 32 22 2F 00<56>
0310:22 21 33 29 23 00 38 2C<1D>
0318:00 00 00 23 6F 6D 70 69<B5>
0320:6C 65 72 00 11 0E 11 00<C6>
0328:00 00 00 08 63 09 00 11<F6>
0330:19 18 15 00 26 72 61 6E<97>
0338:6B 00 2F 73 74 72 6F 77<C4>
0340:73 6B 69 00 60 0D 61 AE<47>
0348:24 E4 AC 25 E4 E8 D0 01<74>
0350:C8 8E 24 21 8C 25 21 A9<F0>
0358:00 8D 0E D4 78 A9 FE 8D<73>
0360:01 D3 A9 CB 8D FA FF A9<60>
0368:20 8D FB FF A9 F0 8D FE<15>
0370:FF A9 20 8D FF FF A9 CC<0E>
0378:20 65 60 A9 E0 20 65 60<B6>
0380:A9 40 8D 0E D4 58 A9 FF<4E>
0388:8D 01 D3 20 8B 60 A9 00<11>
0390:85 09 A5 0C A4 0D 8D 7E<E3>
0398:20 8C 7F 20 A2 01 8E F8<89>
03A0:03 86 09 86 08 CA 8E 44<CB>
03A8:02 4C 80 20 85 D5 A0 00<4D>
03B0:84 D4 A2 04 A9 FF 8D 01<C5>
03B8:D3 B1 D4 99 00 05 C8 D0<CD>
03C0:F8 CE 01 D3 B9 00 05 91<24>
03C8:D4 C8 D0 F8 E6 D5 CA D0<0A>
03D0:E3 60 A2 10 A9 D6 9D 44<D1>
03D8:03 A9 00 9D 45 03 9D 49<0E>
03E0:03 A9 04 9D 48 03 A9 07<93>
03E8:9D 42 03 20 56 E4 30 5A<E4>
03F0:A9 0E 85 DA 9D 44 03 A9<C2>
03F8:61 85 DB 9D 45 03 38 A5<BF>
0400:D8 E5 D6 85 DC A5 D9 E5<C2>
0408:D7 85 DD E6 DC D0 02 E6<1F>
0410:DD A5 DC 9D 48 03 A5 DD<A9>
0418:9D 49 03 20 56 E4 30 32<36>
0420:CE 01 D3 A0 00 A6 DD F0<E5>
0428:0E B1 DA 91 D6 C8 D0 F9<2E>
0430:E6 D7 E6 DB CA D0 F2 A6<82>
0438:DC F0 08 B1 DA 91 D6 C8<BE>
0440:CA D0 F8 A9 FF 8D 01 D3<B9>
0448:30 88 A9 0C 9D 42 03 4C<78>
0450:56 E4 6C 0A 00 E2 02 E3<B8>
0458:02 00 60 00 C0 3D CB 20<13>
0460:CA C0 18 A5 58 69 78 8D<9F>
```

Listing 1. Turbo-Basic-Compiler (Länge: 9933 Byte). Geben Sie das Programm bitte mit AMPEL von Seite 87 ein.

```
0468:44 21 A5 59 69 00 8D 45<1D>
0470:21 A9 38 8D 30 02 A9 21<C0>
0478:8D 31 02 A9 03 85 54 20<61>
0480:1C C1 A2 FF 9A 20 BE F1<CA>
0488:20 ED C2 20 54 C2 20 5C<57>
0490:C1 4C 2F C0 20 EF F8 20<0A>
0498:F1 F4 20 EA E4 20 EA E4<14>
04A0:A5 90 05 91 F0 0A A5 90<8B>
04A8:A4 91 20 E8 C1 4C 5A C0<19>
04B0:20 EF E4 4B 65 69 6E 65<C0>
04B8:00 20 EF E4 20 46 65 68<31>
04C0:6C 65 72 9B 50 72 6F 67<86>
04C8:72 61 6D 6D 6C 61 65 6E<6E>
04D0:67 65 20 3D 20 00 38 A5<53>
04D8:88 E9 6F AA A5 89 E9 22<86>
04E0:A8 8A 20 E8 C1 20 EF E4<CF>
04E8:20 42 79 74 65 73 9B 00<8B>
04F0:A5 90 05 91 D0 06 20 47<4D>
04F8:F7 4C 23 C0 A2 10 20 83<4F>
0500:C4 A9 00 85 A2 20 EF E4<3A>
0508:9B 9B 42 69 74 74 65 20<AD>
0510:65 69 6E 65 20 54 61 73<D1>
0518:74 65 20 64 72 0A 63 6B<E3>
0520:65 6E 00 20 F7 C0 4C 23<EC>
0528:C0 A2 00 20 83 C4 A9 C0<F5>
0530:85 6A A9 03 9D 42 03 A9<39>
0538:0C 9D 4A 03 A9 00 9D 4B<FC>
0540:03 A9 AC 9D 44 03 A9 20<14>
0548:9D 45 03 20 AF 20 30 D9<83>
0550:A9 CC 8D F4 02 60 86 A5<D2>
0558:84 A6 20 1E 21 C0 00 30<AF>
0560:05 A6 A5 A4 A6 60 20 00<12>
0568:E4 53 74 6F 70 70 65 64<08>
0570:00 4C 9D C0 A0 FD A2 00<21>
0578:4C FE 20 A0 27 A9 00 99<9B>
0580:99 21 88 10 FA 60 8A 48<3F>
0588:98 48 A2 10 A9 07 9D 42<02>
0590:03 A9 00 9D 48 03 9D 49<E2>
0598:03 20 88 C4 85 9D E6 95<7A>
05A0:68 A8 68 AA A5 9D 60 86<56>
05A8:9D A6 97 E6 97 BD 80 04<AA>
05B0:A6 9D 85 9D 60 A2 FF 9A<78>
05B8:4C 35 C0 A9 00 85 95 20<19>
05C0:27 C1 85 92 20 27 C1 85<FC>
05C8:93 A8 30 E9 20 27 C1 85<3C>
05D0:94 A9 04 A2 88 20 FF C3<EF>
05D8:A5 92 91 AB C8 A5 93 91<E1>
05E0:AB C8 A5 80 91 AB C8 A5<74>
05E8:81 91 AB A5 92 A4 93 20<59>
05F0:2C C2 A9 00 85 8F 20 AE<82>
05F8:C4 A5 95 C5 94 90 F7 A5<EE>
0600:8F F0 19 A0 01 B1 8E 48<2E>
0608:A5 80 91 8E C8 B1 8E AA<9D>
0610:A5 81 91 8E 86 8F 68 85<CE>
0618:8E 4C A0 C1 60 85 D4 84<CB>
0620:D5 A9 00 85 D6 85 D7 85<14>
0628:D8 A2 10 F8 06 D4 26 D5<AA>
0630:A5 D6 65 D6 85 D6 A5 D7<B9>
0638:65 D7 85 D7 A5 D8 65 D8<F8>
0640:85 D8 CA D0 E7 D8 60 20<50>
0648:BE C1 A9 00 85 D9 A5 D8<4B>
0650:20 0C C2 A5 D7 20 03 C2<F1>
0658:A5 D6 20 03 C2 A5 D9 F0<AA>
0660:12 60 48 4A 4A 4A 4A 20<2B>
0668:0C C2 68 29 0F D0 04 A4<EE>
0670:D9 F0 17 09 30 85 D9 48<E9>
0678:8A 48 98 48 BA BC 03 01<07>
0680:A6 A2 20 FE 20 68 A8 68<86>
0688:AA 68 60 20 BE C1 A0 16<11>
0690:A5 D8 20 46 C2 A5 D7 20<3C>
0698:3D C2 A5 D6 48 4A 4A 4A<27>
06A0:4A 20 46 C2 68 29 0F 09<B5>
06A8:10 99 99 21 C8 60 3A 65<C7>
06B0:69 6C 65 A0 04 B9 4F C2<54>
06B8:99 A9 21 88 10 F7 A9 07<CE>
06C0:A2 10 9D 42 03 A9 0E 9D<ED>
06C8:48 03 A9 00 9D 49 03 A9<83>
06D0:D4 9D 44 03 A9 00 9D 45<AC>
06D8:03 20 88 C4 4C 80 C2 A5<46>
06E0:D4 05 D5 F0 1A 20 03 E4<ED>
06E8:4B 65 69 6E 20 42 61 73<9D>
06F0:69 63 2D 50 72 6F 67 72<F0>
06F8:61 6D 6D 00 4C 9D C0 A5<78>
0700:D6 C5 DA A5 D7 E5 DB B0<E8>
0708:0B 20 27 C1 E6 D6 D0 EF<A9>
0710:E6 D7 D0 EB A5 DA C5 DC<73>
0718:A5 DB E5 DD B0 2D 20 27<B7>
0720:C1 A8 20 27 C1 AA 98 09<18>
0728:3F 9D 00 DB 20 27 C1 20<26>
0730:27 C1 20 27 C1 20 27 C1<16>
0738:20 27 C1 20 27 C1 18 A5<6B>
0740:DA 69 08 85 DA 90 CD E6<3F>
```

```
0748:DB B0 C9 60 A2 80 A9 6F<E0>
0750:95 00 A9 22 95 01 E8 E8<60>
0758:E0 8C 90 F2 A9 FF A0 00<13>
0760:84 90 84 91 84 8C 84 8D<7F>
0768:85 92 85 93 99 00 DC 99<EC>
0770:00 DD 99 00 DB C8 D0 F4<5A>
0778:20 D5 EC A9 FF 91 AB 60<C5>
0780:07 C5 AD EE DD EF 4A EA<F9>
0788:2B E4 2B E4 69 EA 12 EE<D1>
0790:F4 EE 32 EF 1A EA 1A EA<DF>
0798:1A EA 52 F0 7C E9 2B E4<31>
07A0:5C EA 4E EC 7F E9 67 E9<1D>
07A8:5C EA 82 E9 2B E4 FE F0<F3>
07B0:2B E4 2B E4 94 EF BC EF<B7>
07B8:CF EF F3 F0 17 F1 63 EB<73>
07C0:7C EA 64 E9 ED EF 48 F0<C5>
07C8:79 E9 6B EC 82 E9 73 E9<FF>
07D0:7C EA 8A EF 63 EF 63 EC<B8>
07D8:F6 EB F9 EB 82 E9 F3 EB<F3>
07E0:2E F0 B2 EF 12 F0 74 EA<E9>
07E8:2B E4 2B E4 69 EA 2B E4<D5>
07F0:99 EB BD EC C0 EC 01 EA<16>
07F8:DC EC 60 ED B9 ED 04 EE<BD>
0800:88 EE 62 EE AD EC AA EC<51>
0808:F0 EB DF EC 66 ED D7 ED<E0>
0810:C5 F0 D4 F0 D7 F0 DA F0<5A>
0818:DD F0 7A F0 E0 F0 8A F0<8A>
0820:98 F0 76 E9 52 EA 2B E4<24>
0828:07 C5 2B E4 2B E4 2B E4<B3>
0830:2B E4 D8 EE E3 F0 E6 F0<43>
0838:9D F0 8A F0 04 EA 43 EC<FD>
0840:7C EC 15 F0 22 EC 75 EF<43>
0848:97 EF 20 00 E4 53 70 65<F3>
0850:69 63 68 65 72 20 76 6F<30>
0858:6C 6C 00 4C 9D C0 85 A4<8E>
0860:18 65 8A A8 A5 8B 69 00<6B>
0868:CC E5 02 ED E6 02 B0 DA<17>
0870:38 A5 8A F5 00 85 A9 A5<70>
0878:8B F5 01 85 AA 18 B5 00<0B>
0880:85 A5 85 AB 65 A4 85 A7<59>
0888:B5 01 85 A6 85 AC 69 00<52>
0890:85 A8 B5 00 65 A4 95 00<C6>
0898:B5 01 69 00 95 01 E8 E8<D8>
08A0:E0 8C 90 EE A6 AA 18 8A<02>
08A8:65 A6 85 A6 18 8A 65 A8<25>
08B0:85 A8 E8 A4 A9 F0 25 88<3B>
08B8:B1 A5 91 A7 98 D0 F8 F0<C2>
08C0:1B C6 A6 C6 A8 88 B1 A5<4C>
08C8:91 A7 88 B1 A5 91 A7 88<29>
08D0:B1 A5 91 A7 88 B1 A5 91<EA>
08D8:A7 98 D0 E9 CA D0 E2 A0<BC>
08E0:00 60 A9 0C 9D 42 03 20<CE>
08E8:AF 20 C0 00 30 01 60 98<2C>
08F0:48 20 EF E4 49 2F 4F 20<0D>
08F8:45 72 72 6F 72 20 2D 20<D6>
0900:00 A0 00 68 20 E8 C1 20<F2>
0908:EA E4 4C 9D C0 20 27 C1<F7>
0910:85 96 A0 00 84 9A 84 A3<8C>
0918:20 27 C1 99 80 04 C8 A6<C7>
0920:95 E4 96 90 F3 C9 14 D0<54>
0928:05 A9 16 99 7F 04 99 80<F0>
0930:04 C8 D0 FA C8 84 97 AD<40>
0938:81 04 85 9D AD 80 04 85<F6>
0940:99 0A A8 B9 21 C3 85 D4<B2>
0948:B9 22 C3 85 D5 20 08 C5<EC>
0950:A5 9A F0 12 A2 80 20 FF<8C>
0958:C3 A6 9A B9 00 D9 91 AB<67>
0960:C8 CA D0 F7 86 9A 60 6C<F9>
0968:D4 00 20 41 C5 C6 9B C6<B4>
0970:9B F0 20 A4 9B B9 01 D8<34>
0978:48 B9 00 D8 18 69 08 48<14>
0980:A9 8F 20 F3 E9 20 45 C5<55>
0988:68 20 F3 E9 68 20 F3 E9<56>
0990:4C 31 C9 AD 01 D8 48 A5<95>
0998:9D 18 69 73 48 4C 26 C5<38>
09A0:A0 00 84 9B A0 00 84 9C<6C>
09A8:84 A0 A9 11 8D 00 DA 20<6B>
09B0:DD C5 90 FB 85 9E AA BD<D3>
09B8:C9 C6 85 9F A4 9C BE 00<56>
09C0:DA BD 27 C7 C5 9F 90 1C<24>
09C8:AA D0 01 60 B9 00 DA C6<11>
09D0:9C A8 0A AA BD FD C5 85<60>
09D8:D4 BD FE C5 85 D5 20 08<DB>
09E0:C5 4C 5D C5 A5 9E C9 3C<D3>
09E8:D0 2E B9 00 DA 88 C9 58<60>
09F0:F0 22 C9 6B F0 1E C9 38<64>
09F8:F0 10 C9 37 F0 0C C9 39<1E>
0A00:F0 08 C9 3B F0 04 C9 3F<3C>
0A08:D0 E0 A9 83 20 F3 E9 A9<D1>
0A10:8F 20 F3 E9 A5 9E A4 9C<69>
0A18:C9 37 90 17 C9 3C B0 13<6B>
0A20:C9 3A F0 0F A6 9B BD FF<26>
```

```
0A28:D7 C6 9B C6 9B C8 99 00<F7>
0A30:DA A5 9E C8 99 00 DA 84<8F>
0A38:9C 4C 50 C5 20 48 C1 09<30>
0A40:00 30 2E F0 29 C9 0F 90<3F>
0A48:03 F0 0D 60 A0 80 20 CF<83>
0A50:C7 20 26 C6 A0 06 D0 0E<BD>
0A58:A0 81 20 CC C7 20 26 C6<D4>
0A60:20 48 C1 A8 F0 06 20 48<C8>
0A68:C1 88 D0 FA 18 60 20 48<63>
0A70:C1 49 80 48 A0 82 20 CF<42>
0A78:C7 68 A8 BE 00 DB 86 98<4B>
0A80:20 F3 E9 18 60 18 A9 80<44>
0A88:65 97 08 20 F3 E9 28 A9<74>
0A90:04 69 00 4C F3 E9 C4 C7<0F>
0A98:C4 C7 C4 C7 C4 C7 C4 C7<CD>
0AA0:C4 C7 BD C7 BD C7 BD C7<AD>
0AA8:BD C7 BD C7 01 C8 C4 C7<4A>
0AB0:C4 C7 22 C8 C9 C8 00 00<86>
0AB8:00 00 C4 C7 C4 C7 C4 C7<41>
0AC0:C4 C7 C4 C7 C4 C7 22 C8<A2>
0AC8:9B C7 49 C8 4C C8 46 C8<37>
0AD0:C9 C8 46 C8 C7 C8 95 C7<EE>
0AD8:F5 C7 0B C8 A6 C7 9B C7<3F>
0AE0:A6 C7 A6 C7 9B C7 9B C7<CF>
0AE8:A1 C7 9B C7 9B C7 A6 C7<11>
0AF0:9B C7 9B C7 9B C7 9B C7<53>
0AF8:9B C7 9B C7 9B C7 B0 C7<85>
0B00:B0 C7 B0 C7 B0 C7 A1 C7<47>
0B08:A1 C7 A1 C7 23 C8 CC C7<D8>
0B10:A1 C7 EE C7 A6 C7 BD C7<5F>
0B18:9B C7 CC C7 CF C7 BD C7<65>
0B20:00 00 C2 C8 9B C7 9B C7<DF>
0B28:89 C8 89 C8 89 C8 89 C8<86>
0B30:00 00 23 C8 CF C7 CF C7<1D>
0B38:00 00 00 00 00 00 00 00<4E>
0B40:00 00 00 00 00 20 20 20<37>
0B48:20 20 20 2C 28 22 22 28<73>
0B50:1E 1A 1C 32 04 00 00 30<E2>
0B58:30 30 30 30 30 2E 2E 32<E3>
0B60:32 32 32 32 04 32 32 32<62>
0B68:32 32 32 32 32 32 32 32<FD>
0B70:32 32 32 32 32 32 32 32<84>
0B78:32 32 32 32 32 32 2A 2A<74>
0B80:32 32 2A 32 32 28 32 32<4D>
0B88:32 28 00 32 32 32 32 32<95>
0B90:32 32 00 32 32 32 00 00<81>
0B98:00 00 00 00 00 00 00 00<AE>
0BA0:00 00 00 20 20 20 20 20<9A>
0BA8:20 2C 28 22 22 28 1D 1A<F6>
0BB0:1C 02 30 01 01 30 30 30<C0>
0BB8:30 30 30 2D 2D 02 02 02<43>
0BC0:02 02 03 02 02 02 02 02<F6>
0BC8:02 02 02 02 02 02 02 02<DE>
0BD0:02 02 02 02 02 02 02 02<E6>
0BD8:02 02 02 02 2A 2A 02 32<00>
0BE0:2A 02 02 28 02 32 32 28<32>
0BE8:00 32 02 02 32 32 32 32<9C>
0BF0:00 02 32 32 A9 80 85 98<D8>
0BF8:D0 02 46 98 98 4C F3 E9<FE>
0C00:A9 83 20 F3 E9 20 9D C7<8A>
0C08:A9 84 46 98 4C F3 E9 A9<C1>
0C10:83 20 F3 E9 A9 97 20 F3<C5>
0C18:E9 4C A6 C7 46 98 A9 85<3F>
0C20:4C 6A C8 20 BD C7 A9 88<CE>
0C28:4C F3 E9 A9 80 2C A9 00<E0>
0C30:85 98 A6 9B F0 14 BD FF<67>
0C38:D7 C9 8F 90 08 C9 93 90<F3>
0C40:09 C9 95 F0 05 A9 87 20<9C>
0C48:F3 E9 4C 9D C7 A9 83 85<CC>
0C50:98 4C 6A C8 A9 83 85 98<CB>
0C58:20 F3 E9 A9 97 4C 6A C8<27>
0C60:20 9D C7 46 98 A9 88 4C<69>
0C68:F3 E9 A9 83 20 F3 E9 A9<39>
0C70:9F A4 A0 20 6A C8 A9 00<13>
0C78:85 A0 85 98 A9 84 4C F3<64>
0C80:E9 60 46 98 C6 A0 F0 06<FB>
0C88:C6 A0 A9 83 D0 0F A9 87<94>
0C90:20 F3 E9 A9 89 20 F3 E9<C0>
0C98:A9 00 20 F3 E9 20 6A C8<A0>
0CA0:A9 84 4C F3 E9 A9 AB 2C<52>
0CA8:A9 9B 2C A9 9A 18 65 A0<93>
0CB0:65 A0 48 A9 83 20 F3 E9<5A>
0CB8:A6 9C C6 9C BC 00 DA B9<57>
0CC0:00 DB 85 98 68 A2 00 86<FD>
0CC8:A0 20 F3 E9 4C 9D C7 A0<A4>
0CD0:00 84 9B 20 45 C5 4C 31<3E>
0CD8:C9 A0 00 84 9B 20 45 C5<5A>
0CE0:A9 83 20 F3 E9 4C 31 C9<03>
0CE8:98 48 A0 80 20 CF C7 A4<C8>
0CF0:A3 68 38 E9 66 99 81 05<29>
0CF8:F0 02 A9 40 99 80 05 A9<2F>
0D00:00 99 82 05 99 83 05 99<5E>
```

Listing 1. Turbo-Basic-Compiler (Fortsetzung)

```
0D08:84 05 99 85 05 18 98 69<41>
0D10:06 85 A3 69 7A 08 20 F3<60>
0D18:E9 28 A9 05 69 00 4C F3<E4>
0D20:E9 A0 48 4C CF C7 E6 A0<51>
0D28:68 68 A4 9C 4C 6D C5 B9<E1>
0D30:02 D8 99 00 D8 C8 C4 9B<76>
0D38:D0 F5 C6 9B 4C F1 C8 8A<2B>
0D40:99 00 D8 C8 C8 B9 01 D8<2D>
0D48:99 00 D8 C8 C4 9B D0 F5<47>
0D50:C6 9B 4C 38 C9 20 20 CB<09>
0D58:90 23 48 20 00 E4 49 6E<5D>
0D60:74 65 67 65 72 20 3C 30<3D>
0D68:20 6F 64 65 72 20 3E 36<07>
0D70:35 35 33 35 00 68 A8 A9<C6>
0D78:00 85 D4 85 D5 A9 89 99<AC>
0D80:00 D8 A5 D4 99 01 D8 A5<87>
0D88:D5 99 02 D8 C8 4C E4 C8<34>
0D90:A9 00 20 F3 E9 C6 9B A0<C6>
0D98:00 B9 00 D8 10 67 C9 BB<D5>
0DA0:B0 62 C9 AF F0 5C B0 5B<E8>
0DA8:C9 96 F0 58 C9 81 F0 53<83>
0DB0:C9 8C F0 4F C9 8E F0 4B<AC>
0DB8:C9 94 F0 47 BE 01 D8 C9<F9>
0DC0:9A F0 58 C9 9C F0 57 C9<51>
0DC8:98 B0 39 C9 80 F0 3C C9<39>
0DD0:82 F0 4E C9 83 F0 56 C9<55>
0DD8:84 F0 59 C9 88 F0 5C C9<EE>
0DE0:89 F0 63 C9 8A F0 1D C9<23>
0DE8:8B F0 2D C9 8D F0 15 C9<0D>
0DF0:86 F0 10 C9 93 F0 0C C9<0D>
0DF8:87 F0 6F C9 85 D0 06 4C<0B>
0E00:3B CA C8 C8 C8 C8 C4 9B<29>
0E08:90 8F 60 B9 03 D8 C9 83<33>
0E10:D0 03 4C F6 C8 4C CB CA<88>
0E18:4C 05 CB A2 A9 2C A2 AA<02>
0E20:2C A2 8A B9 02 D8 C9 83<3B>
0E28:D0 DA 4C E0 C8 E0 84 D0<C2>
0E30:D4 4C D0 C8 E0 83 F0 F9<71>
0E38:4C 5E CA E0 83 F0 04 E0<CC>
0E40:84 D0 C2 4C E6 C8 B9 03<D1>
0E48:D8 C9 46 F0 0B C9 97 F0<E2>
0E50:0F C9 55 D0 AE A9 93 2C<3F>
0E58:A9 86 99 00 D8 4C 2D C9<B0>
0E60:A9 96 99 00 D8 C8 C8 4C<46>
0E68:D0 C8 E0 80 F0 0E E0 82<CC>
0E70:F0 19 E0 89 D0 03 4C A8<28>
0E78:CA 4C A6 C9 B9 04 D8 C9<6C>
0E80:85 D0 F6 A9 8B 20 49 CA<DC>
0E88:4C 2D C9 B9 03 D8 C9 85<4A>
0E90:D0 E7 A9 8D 20 49 CA 4C<43>
0E98:E4 C8 8A 38 E9 24 C9 04<BA>
0EA0:B0 D7 99 00 D8 4C E5 C8<FF>
0EA8:99 00 D8 98 48 B9 02 D8<86>
0EB0:99 01 D8 C8 C4 9B D0 F5<F6>
0EB8:C6 9B 68 A8 60 E0 A0 F0<71>
0EC0:13 E0 A2 F0 12 E0 A3 F0<D6>
0EC8:11 E0 B8 F0 15 E0 B9 F0<09>
0ED0:11 4C A6 C9 A9 A6 2C A9<8E>
0ED8:A7 2C A9 A8 99 00 D8 4C<AB>
0EE0:E5 C8 B9 04 D8 C9 83 D0<99>
0EE8:E8 C8 20 20 CB 88 B0 E1<1D>
0EF0:A5 D5 D0 DD 18 B9 01 D8<5D>
0EF8:69 03 99 00 D8 A5 D4 99<50>
0F00:01 D8 C8 C8 4C F2 CA B9<82>
0F08:04 D8 C9 56 F0 0B C9 57<32>
0F10:F0 0A C9 5A D0 BB A9 94<B6>
0F18:2C A9 8C 2C A9 8E 99 01<9E>
0F20:D8 A9 83 99 00 D8 C8 4C<79>
0F28:2D C9 C9 49 F0 23 C9 48<A2>
0F30:F0 1F C9 A0 F0 41 C9 36<6D>
0F38:D0 27 B9 02 D8 85 D5 B9<56>
0F40:01 D8 85 D4 A2 00 A1 D4<CC>
0F48:F0 04 49 80 81 D4 4C 2D<83>
0F50:C9 B9 03 D8 99 00 D8 C8<2B>
0F58:C4 9B D0 F5 C6 9B 4C DB<9C>
0F60:C8 4C A4 C9 B9 03 D8 C9<C2>
0F68:1D 90 F6 C9 28 B0 F2 69<D1>
0F70:93 99 00 D8 4C 2D C9 A9<8D>
0F78:AF 99 00 D8 4C 2D C9 98<52>
0F80:48 C8 B9 00 D8 85 E0 C8<64>
0F88:B9 00 D8 85 E1 A0 05 B1<AA>
0F90:E0 99 D4 00 88 10 F8 20<EB>
0F98:2A 21 68 A8 60 00 E4 27<7D>
0FA0:F9 20 DA E4 E6 90 D0 02<AD>
0FA8:E6 91 68 85 D4 68 85 D5<31>
0FB0:E6 D4 D0 02 E6 D5 A0 00<D6>
0FB8:B1 D4 F0 06 20 18 C2 4C<B3>
0FC0:0F E4 20 EA E4 A5 D5 48<89>
0FC8:A5 D4 48 60 20 DA E4 A2<0F>
0FD0:00 BD 8A E4 85 A4 F0 0D<2D>
0FD8:C5 99 F0 09 E8 BD 8A E4<79>
0FE0:10 FA E8 D0 EC E8 BD 8A<13>
```

```
0FE8:E4 29 7F 20 18 C2 BD 8A<41>
0FF0:E4 10 F2 A5 A4 F0 18 20<C9>
0FF8:03 E4 20 6E 69 63 68 74<95>
1000:20 63 6F 6D 70 69 6C 69<F0>
1008:65 72 62 61 72 00 60 A5<B1>
1010:99 A0 00 20 E8 C1 20 03<6C>
1018:E4 20 2A 2A 2A 20 4D 55<D9>
1020:45 4C 4C 20 2A 2A 2A 00<BF>
1028:4C 9D C0 04 4C 49 53 D4<C7>
1030:05 45 4E 54 45 D2 0F 43<F5>
1038:4F 4E D4 16 4E 45 D7 18<74>
1040:4C 4F 41 C4 19 53 41 56<18>
1048:C5 34 43 53 41 56 C5 35<73>
1050:43 4C 4F 41 C4 37 45 52<68>
1058:52 4F D2 53 2A CC 55 52<D1>
1060:45 4E 55 CD 56 44 45 CC<2E>
1068:57 44 55 4D D0 58 54 52<4A>
1070:41 43 C5 00 54 4F 4B 45<35>
1078:4E 20 A3 20 15 C1 A5 92<84>
1080:A4 93 20 E8 C1 A9 3A 2C<67>
1088:A9 20 2C A9 9B 4C 18 C2<4D>
1090:68 85 D4 68 85 D5 E6 D4<1E>
1098:D0 02 E6 D5 A0 00 B1 D4<7F>
10A0:F0 06 20 18 C2 4C F5 E4<CC>
10A8:A5 D5 48 A5 D4 48 60 98<BA>
10B0:48 8A A0 00 20 E8 C1 20<91>
10B8:00 E4 51 4A 53 42 00 68<98>
10C0:A8 60 0A AA A9 20 20 A9<1F>
10C8:E9 BD 00 DF 20 A9 E9 BD<AE>
10D0:01 DF F0 DB 4C A9 E9 20<CB>
10D8:00 E4 43 6F 6D 70 69 6C<AD>
10E0:65 72 20 46 65 68 6C 65<76>
10E8:72 00 4C 9D C0 A4 9B C0<16>
10F0:07 D0 25 AD 00 D8 C9 82<6B>
10F8:D0 1E AD 02 D8 C9 B8 D0<82>
1100:17 AD 05 D8 C9 A0 D0 10<6F>
1108:AD 01 D8 CD 06 D8 D0 08<E1>
1110:8D 05 D8 A0 02 4C C6 E5<ED>
1118:A0 00 C4 9B F0 41 B9 00<2E>
1120:D8 AA BD CE E7 10 30 C9<BD>
1128:C0 90 1B C9 D7 B0 A8 29<FF>
1130:3F 0A AA BD CE E8 85 D4<60>
1138:BD CF E8 85 D5 B9 00 D8<17>
1140:20 08 C5 4C B9 E5 A9 A2<16>
1148:20 A9 E9 C8 B9 00 D8 20<E5>
1150:A9 E9 BD CE E7 29 3F 20<0F>
1158:21 E5 C8 C4 9B D0 BF 60<12>
1160:38 E9 50 2C A9 80 2C A9<CA>
1168:81 2C A9 05 2C A9 04 85<50>
1170:E8 C8 B9 00 D8 85 E6 C8<34>
1178:B9 00 D8 85 E7 84 E9 A0<83>
1180:05 B1 E6 99 E0 00 88 10<B7>
1188:F8 A5 E0 D0 06 A5 E8 C9<28>
1190:04 F0 10 A5 E5 05 E4 05<15>
1198:E3 05 E2 D0 34 A5 E0 C9<5F>
11A0:40 D0 2E A5 E8 A4 E9 A6<2C>
11A8:E1 C9 66 F0 22 C9 67 F0<2A>
11B0:36 C9 68 F0 4C C9 69 F0<72>
11B8:4F C9 6A F0 32 C9 04 F0<4D>
11C0:36 C9 05 F0 46 C9 80 F0<1C>
11C8:51 C9 81 F0 4A D0 54 E0<8E>
11D0:02 D0 50 A9 71 2C A9 70<82>
11D8:2C A9 6F 2C A9 72 2C A9<9B>
11E0:43 2C A9 42 4C 21 E5 E0<70>
11E8:02 F0 EB A9 73 D0 1E E0<EC>
11F0:02 F0 E9 A9 76 D0 16 E0<AD>
11F8:01 F0 E4 90 E5 A9 40 D0<34>
1200:0C E0 01 F0 D4 A9 74 2C<9F>
1208:A9 75 2C A9 79 48 A9 A9<BD>
1210:20 36 EB 68 4C 21 E5 A9<F7>
1218:78 2C A9 77 48 20 02 E9<97>
1220:4C 6D E6 A5 82 85 D4 C5<2B>
1228:84 A5 83 85 D5 E5 85 B0<0D>
1230:21 A0 05 B1 D4 D9 E0 00<42>
1238:D0 05 88 10 F6 30 24 18<64>
1240:A5 D4 69 06 85 D4 90 02<D7>
1248:E6 D5 C5 84 A5 D5 E5 85<2C>
1250:90 DF A9 06 A2 84 20 FF<CB>
1258:C3 A0 05 B9 E0 00 91 D4<E6>
1260:88 10 F8 A9 0F 20 A9 E9<89>
1268:38 A5 D4 E5 82 08 20 A9<B8>
1270:E9 A9 A0 20 A9 E9 28 A5<4F>
1278:D5 E5 83 20 A9 E9 A4 E9<D3>
1280:A5 E8 10 0B 20 02 E9 A9<A7>
1288:3B C6 E8 10 02 A9 3F 4C<77>
1290:21 E5 A2 46 20 FD E6 A9<B9>
1298:A0 A2 00 4C 36 EB C8 B9<9F>
12A0:00 D8 85 D4 C8 B9 00 D8<C9>
12A8:85 D5 C9 C0 90 0C C9 D8<99>
12B0:B0 20 C9 D3 F0 1C C9 D0<4C>
12B8:90 18 A9 A5 2C A9 A4 A6<1D>
12C0:D5 F0 0A 09 08 20 2C E7<1B>
```

```
12C8:A5 D5 4C A9 E9 A6 D4 4C<62>
12D0:36 EB 68 68 8A 48 88 88<E2>
12D8:20 FC E8 68 4C 84 E9 A2<EF>
12E0:55 20 FD E6 E6 D4 D0 D5<A7>
12E8:E6 D5 4C 1C E7 A9 17 20<26>
12F0:A9 E9 38 A5 86 E5 84 08<45>
12F8:20 A9 E9 A9 A0 20 A9 E9<C5>
1300:28 A5 87 E5 85 20 A9 E9<89>
1308:C8 B9 00 D8 85 E0 C8 B9<BA>
1310:00 D8 85 E1 98 48 A0 00<0C>
1318:B1 E0 18 69 01 85 E2 A2<AC>
1320:86 20 FF C3 A0 00 B1 E0<6D>
1328:91 AB C8 C4 E2 D0 F7 68<3A>
1330:A8 A9 1B 4C 21 E5 A2 55<B1>
1338:2C A2 88 2C A2 8A 2C A2<B8>
1340:8C 20 A7 E7 E8 A9 84 2C<8C>
1348:A9 85 4C 36 EB A2 09 2C<20>
1350:A2 29 2C A2 49 20 08 E9<F3>
1358:A9 AA 20 A9 E9 A9 98 20<35>
1360:A9 E9 20 08 E9 A9 A8 20<75>
1368:A9 E9 A9 8A 4C A9 E9 00<32>
1370:01 02 03 FF FF FF FF FF<D8>
1378:FF FF FF FF FF FF FF FF<9E>
1380:FF FF FF FF FF FF FF FF<A6>
1388:FF FF FF FF 1D 1E 1F 20<7D>
1390:21 22 23 24 25 26 27 28<AA>
1398:29 2A FF FF FF FF 2F 30<2F>
13A0:31 32 33 34 FF 36 FF FF<63>
13A8:FF FF FF FF 3D 3E FF D3<8D>
13B0:41 D0 D1 44 45 46 47 48<DF>
13B8:49 4A 4B 4C 4D 4E 4F 50<45>
13C0:51 52 53 54 55 56 57 58<4D>
13C8:59 5A 5B 5C 5D 5E 5F 35<2A>
13D0:37 FF FF 38 39 FF FF FF<F7>
13D8:FF FF 3A 6B 6C FF FF FF<78>
13E0:FF FF FF FF FF FF FF FF<06>
13E8:FF FF FF FF FF FF FF C0<CF>
13F0:C2 86 2C 2D 2B C4 0D CD<70>
13F8:C3 88 C1 C6 87 C7 2E C9<E9>
1400:CA CB C5 C8 CC D2 CD FF<53>
1408:FF 89 8A 8B 8C FF 8E 8F<C0>
1410:90 91 92 93 94 95 96 97<A5>
1418:98 99 D4 FF 9C FF CF CE<21>
1420:CE CE CE CE CE CE CE CE<70>
1428:CE CE D5 D6 FF FF FF FF<93>
1430:FF FF FF FF FF FF FF FF<58>
1438:FF FF FF FF FF FF FF FF<60>
1440:FF FF FF FF FF FF FF FF<68>
1448:FF FF FF FF FF FF FF FF<70>
1450:FF FF FF FF FF FF FF FF<78>
1458:FF FF FF FF FF FF FF FF<C0>
1460:FF FF FF FF FF FF FF FF<88>
1468:FF FF FF FF FF FF FF CC<5D>
1470:E5 C9 E5 4C E7 FC E8 F1<C8>
1478:E6 3E E7 AF E7 AC E7 B2<60>
1480:E7 98 E7 9B E7 9E E7 95<C1>
1488:E7 BE E5 BF E5 C3 E5 1E<46>
1490:E9 21 E9 05 E9 2E E9 38<EC>
1498:E9 4E E9 51 E9 20 05 E9<75>
14A0:A9 A0 2C A9 A2 2C A9 A9<CD>
14A8:24 8A 20 A9 E9 C8 B9 00<AD>
14B0:D8 4C A9 E9 A9 07 20 09<26>
14B8:E9 4C 0C E9 A2 BA 2C A2<6E>
14C0:D6 2C A2 D4 A9 A5 20 36<A6>
14C8:EB E8 A9 A4 4C 36 EB 20<08>
14D0:F6 E6 A9 B1 A2 D4 4C 36<C6>
14D8:EB A9 A2 C8 BE 00 D8 20<B6>
14E0:36 EB BD 00 DB 10 03 A9<3B>
14E8:3C 2C A9 1A 4C 21 E5 A9<41>
14F0:6D 2C A9 6E 48 A9 A2 20<0B>
14F8:A9 E9 C8 B9 00 D8 20 A9<FC>
1500:E9 68 4C 21 E5 A0 00 2C<05>
1508:A0 06 A9 A9 20 A5 E9 A0<4C>
1510:FB 4C 3D EB A9 1F 2C A9<AC>
1518:1D 2C A9 1C 2C A9 03 2C<EC>
1520:A9 02 2C A9 00 0A AA BC<56>
1528:01 DE D0 10 8A 20 E8 C1<A4>
1530:20 E7 E4 20 00 E4 51 4A<1E>
1538:53 41 00 60 BD 00 DE AA<25>
1540:A9 20 20 A9 E9 8A 20 A9<6F>
1548:E9 98 86 A4 A6 9A 9D 00<A8>
1550:D9 A6 A4 E6 9A F0 1B 60<4D>
1558:20 00 E4 41 75 73 64 72<C3>
1560:75 63 6B 20 7A 75 20 6B<BA>
1568:6F 6D 70 6C 65 78 00 4C<D3>
1570:9D C0 98 48 8A 48 A9 80<09>
1578:A2 80 20 FF C3 A0 00 B9<15>
1580:00 D9 91 AB B9 80 D9 99<DC>
1588:00 D9 C8 10 F2 84 9A 68<1D>
1590:AA 68 A8 60 86 A4 A6 9B<59>
1598:9D 00 D8 A6 A4 E6 9B F0<40>
15A0:B7 60 A0 96 2C A0 95 A9<51>
```

Listing 1. Turbo-Basic-Compiler (Fortsetzung)



```
15A8:A9 20 A9 E9 AD 81 04 49<8F>
15B0:26 F0 02 A9 FF 20 A9 E9<67>
15B8:4C 3D EB 20 7A C8 A4 9B<06>
15C0:C0 03 D0 17 AD 00 D8 C9<D5>
15C8:89 D0 10 A5 99 C9 0C D0<4B>
15D0:05 A9 08 20 84 E9 A0 00<B3>
15D8:4C 13 E9 20 4C E5 A9 09<5E>
15E0:A4 99 C0 0C D0 02 A9 58<8B>
15E8:4C 84 E9 20 E5 EB A0 C8<1A>
15F0:4C 3D EB 20 E5 EB A2 02<07>
15F8:A0 FD 4C 2D EB 20 70 C8<8C>
1600:20 4C E5 A5 9D C9 12 F0<10>
1608:F4 60 20 0B C5 4C 4C E5<8F>
1610:A9 12 4C 84 E9 A9 0C 20<F8>
1618:84 E9 4C 8B EA A5 9D C9<58>
1620:16 F0 ED 20 84 EC A4 97<A5>
1628:C0 01 D0 09 A5 9D C9 12<3E>
1630:D0 03 20 48 C1 A5 9D C9<2A>
1638:16 F0 64 C9 12 F0 71 C9<80>
1640:15 F0 72 20 70 C8 C9 3E<95>
1648:D0 07 20 A8 F1 A9 2B D0<E1>
1650:48 C9 54 D0 07 20 A8 F1<D4>
1658:A9 56 D0 3D A4 9B C0 03<C8>
1660:D0 2C AD 00 D8 C9 81 D0<4C>
1668:25 AD 01 D8 85 D4 AD 02<D2>
1670:D8 85 D5 A0 00 B1 D4 C9<C1>
1678:01 D0 0D C8 B1 D4 AA A9<0A>
1680:A9 20 36 EB A9 2B D0 11<1D>
1688:20 B0 F1 4C 94 EA 20 4C<94>
1690:E5 A9 0D 24 98 30 02 A9<67>
1698:0E 20 84 E9 4C 94 EA A9<04>
16A0:10 20 84 E9 A5 99 C9 33<2B>
16A8:D0 05 A9 11 20 84 E9 60<69>
16B0:A9 0F 20 84 E9 46 9D A4<6E>
16B8:97 B9 80 04 C9 16 F0 E4<F1>
16C0:C9 15 F0 04 C9 12 D0 03<06>
16C8:20 48 C1 4C 94 EA 8A F0<AE>
16D0:0D A9 8D 20 A9 E9 98 20<53>
16D8:A9 E9 8A 4C A9 E9 A9 85<D0>
16E0:4C A5 E9 A9 95 2C A9 90<3F>
16E8:48 20 7A C8 68 20 F3 E9<5E>
16F0:20 7E C8 A4 9B C0 07 D0<80>
16F8:0A AD 00 D8 C9 89 D0 03<E2>
1700:CD 04 D8 60 20 45 EB D0<4E>
1708:29 AD 06 D8 F0 03 20 D8<16>
1710:EB AD 02 D8 C9 C0 90 0C<0D>
1718:C9 D8 B0 16 C9 D3 F0 12<A3>
1720:C9 D0 90 0E A0 04 20 05<F3>
1728:E9 AE 02 D8 AC 01 D8 4C<91>
1730:2D EB 20 4C E5 A9 13 4C<BC>
1738:84 E9 20 45 EB D0 32 AD<62>
1740:02 D8 C9 C0 90 0C C9 D8<B8>
1748:B0 27 C9 D3 F0 23 C9 D0<DC>
1750:90 1F A0 04 20 FC E8 AE<B1>
1758:02 D8 AC 01 D8 20 2D EB<C7>
1760:C8 D0 01 E8 8A F0 05 A9<44>
1768:8C 4C 32 EB A9 84 4C A5<75>
1770:E9 20 4C E5 A9 14 4C 84<85>
1778:E9 20 00 E4 5A 61 68 6C<F4>
1780:3E 32 35 35 00 60 20 7A<6F>
1788:C8 A9 97 20 9B EC 4C 4C<2A>
1790:E5 A9 21 2C A9 19 2C A9<E3>
1798:17 2C A9 15 48 20 42 EB<F6>
17A0:F0 07 20 4C E5 68 4C 84<26>
17A8:E9 A0 00 20 FC E8 A0 04<38>
17B0:20 02 E9 68 18 69 01 20<87>
17B8:84 E9 AD 06 D8 F0 03 20<A5>
17C0:D8 EB 60 20 45 EB A9 91<3B>
17C8:20 EA EB 20 7A C8 A9 92<4A>
17D0:20 EA EB A5 9D C9 16 F0<DC>
17D8:06 20 7A C8 20 4C E5 A9<71>
17E0:27 4C 84 E9 20 45 EB 20<BF>
17E8:4C E5 A9 1B 4C 84 E9 AD<2B>
17F0:81 04 C9 1C D0 09 20 48<E5>
17F8:C1 20 E5 EB A9 07 2C A9<44>
1800:06 4C 84 E9 20 E5 EB A9<38>
1808:1E 4C 84 E9 AD 81 04 C9<99>
1810:16 F0 05 A9 20 4C E8 F0<60>
1818:A9 01 4C 84 E9 20 84 EC<4B>
1820:A9 23 4C 84 E9 A4 97 B9<FB>
1828:80 04 C9 1C D0 09 20 48<1E>
1830:C1 20 E5 EB A9 0A 2C A9<82>
1838:0B 4C 84 E9 20 F3 E9 8A<10>
1840:48 98 48 20 31 C9 68 A8<48>
1848:68 AA 60 A9 24 2C A9 25<65>
1850:48 20 84 EC 20 45 EB 20<09>
1858:4C E5 68 4C 84 E9 A9 04<D6>
1860:2C A9 05 48 20 45 EB A9<88>
1868:91 20 F3 E9 20 7E C8 20<FC>
1870:4C E5 68 4C 84 E9 A9 03<DD>
1878:A2 8A 4C FF C3 A9 02 2C<9E>
1880:A9 01 48 20 D5 EC 68 91<00>
1888:AB C8 A5 80 91 AB C8 A5<24>
1890:81 91 AB 60 38 A5 8A E9<21>
1898:03 85 8A B0 02 C6 8B A0<D0>
18A0:00 B1 8A 60 20 0C ED 20<8F>
18A8:F3 EC F0 F8 60 A0 02 B1<01>
18B0:8A 85 AC 88 B1 8A 85 AB<36>
18B8:18 A5 80 69 03 91 AB C8<10>
18C0:A5 81 69 00 91 AB 60 48<10>
18C8:20 D5 EC 68 AA CA D0 0D<1B>
18D0:20 00 E4 44 4F 20 66 65<EB>
18D8:68 6C 74 00 60 CA D0 11<5C>
18E0:20 00 E4 52 45 50 45 41<04>
18E8:54 20 66 65 68 6C 74 00<43>
18F0:60 20 00 E4 57 48 49 4C<48>
18F8:45 20 66 65 68 6C 74 00<BB>
1900:60 20 8E ED A9 02 2C A9<59>
1908:01 48 20 06 ED 85 D4 68<7E>
1910:C5 D4 D0 B3 C8 B1 8A AA<90>
1918:C8 B1 8A A8 A9 4C 4C A1<B2>
1920:E9 A0 02 B1 8A 85 AC 88<81>
1928:B1 8A 85 AB 4C E9 EC A9<D7>
1930:D0 2C A9 F0 48 20 70 C8<D1>
1938:C9 84 D0 0E 20 A8 F1 A9<F5>
1940:84 A0 D4 20 A5 E9 A9 05<BF>
1948:D0 05 20 4C E5 A9 A5 A0<21>
1950:D4 20 A5 E9 68 A0 03 4C<2B>
1958:A5 E9 A5 80 48 A5 81 48<43>
1960:20 8E ED 20 F1 C4 A9 03<92>
1968:20 FE ED A9 03 20 E1 EC<98>
1970:68 91 AB 68 88 91 AB 60<4A>
1978:A5 8A 85 D4 A5 8B 85 D5<EB>
1980:A0 00 38 A5 D4 E9 03 85<A1>
1988:D4 B0 02 C6 D5 B1 D4 F0<3E>
1990:F1 C9 05 D0 08 A9 1F 20<4A>
1998:84 E9 20 F1 C4 A9 00 20<C2>
19A0:E1 EC 4C 7B ED A9 03 20<CF>
19A8:68 ED 20 F1 C4 20 F3 EC<4D>
19B0:4C 80 ED 20 8E ED A5 9D<58>
19B8:C9 1B F0 13 20 F1 C4 A6<39>
19C0:8C A4 8D 20 7B ED A5 80<A6>
19C8:85 8C A5 81 85 8D 60 CD<D3>
19D0:7F 05 F0 1C 20 7A C8 A4<DD>
19D8:9A A9 F0 99 FE D8 A0 00<B9>
19E0:20 13 E9 A6 95 E4 94 B0<7B>
19E8:06 20 27 C1 4C 42 EE 60<B5>
19F0:20 F1 C4 A6 8E A4 8F 20<83>
19F8:7B ED A5 80 85 8E A5 81<0A>
1A00:85 8F 60 A5 8D F0 3A A2<13>
1A08:00 A0 00 A9 4C 91 8C C8<3E>
1A10:B1 8C 48 18 8A 65 80 91<A2>
1A18:8C C8 B1 8C AA A5 81 69<42>
1A20:00 91 8C 86 8D 68 85 8C<23>
1A28:60 A5 8D F0 14 A0 00 B1<D6>
1A30:8C F0 0E A2 03 20 68 EE<59>
1A38:20 1B EE A9 00 8D 00 D9<66>
1A40:60 20 00 E4 49 46 20 66<72>
1A48:65 68 6C 74 00 60 A5 92<BA>
1A50:A4 93 20 32 EB 20 48 C1<FC>
1A58:20 A9 E9 A5 9D C9 9B D0<4D>
1A60:F4 A5 9A 8D 02 D9 A2 82<30>
1A68:20 FF C3 A6 9A B9 00 D9<B7>
1A70:91 AB C8 CA D0 F7 86 9A<BF>
1A78:60 20 45 EB A9 91 20 9B<A1>
1A80:EC 20 74 C8 24 98 30 05<92>
1A88:A9 3D 20 F3 E9 20 4C E5<D8>
1A90:A9 26 4C 84 E9 20 0B C5<03>
1A98:A4 9B B9 FF D7 48 20 4C<CC>
1AA0:E5 20 70 C8 68 A6 9D E0<D0>
1AA8:1A 08 48 D0 03 20 74 C8<3B>
1AB0:20 4C E5 68 AA A9 A2 20<42>
1AB8:36 EB A9 28 28 F0 02 A9<A7>
1AC0:29 20 84 E9 A9 B0 A2 03<64>
1AC8:20 36 EB 20 F1 C4 A9 05<68>
1AD0:4C FE ED 20 BD F0 20 F1<28>
1AD8:C4 20 06 ED C9 05 D0 13<13>
1AE0:20 0C ED C6 AC A0 FA B1<A9>
1AE8:AB CD 01 D9 D0 05 A9 2A<E0>
1AF0:4C 84 E9 20 D5 EC 20 00<21>
1AF8:E4 46 4F 52 20 66 65 68<86>
1B00:6C 74 00 60 20 84 EC 20<52>
1B08:E5 EB A9 2B 20 84 E9 A5<BD>
1B10:9D C9 16 D0 F2 60 20 84<5B>
1B18:EC 20 70 C8 20 4C E5 A9<77>
1B20:2C 20 84 E9 A5 9D C9 16<B6>
1B28:D0 EF 60 AC 81 04 C0 1C<80>
1B30:D0 0E A9 2D 2C A9 4A 2C<7C>
1B38:A9 2E 48 20 84 EC 68 2C<29>
1B40:A9 4B 48 20 BD F0 68 48<4F>
1B48:20 84 E9 A5 9D C9 16 D0<9A>
1B50:F2 68 60 20 42 EB 20 4C<09>
1B58:E5 A9 2F D0 E5 20 84 EC<FB>
1B60:20 BD F0 20 A6 F0 A9 A0<B4>
1B68:20 A5 E9 A9 30 4C 84 E9<2C>
1B70:20 84 EC 20 45 EB 20 4C<71>
1B78:E5 A9 31 4C 84 E9 20 84<B8>
1B80:EC AD 81 04 C9 0F F0 1D<AF>
1B88:A9 32 2C A9 36 2C A9 34<7F>
1B90:48 20 BD F0 68 48 06 98<91>
1B98:69 00 20 84 E9 A5 9D C9<2B>
1BA0:16 D0 EE 68 60 20 70 C8<44>
1BA8:20 B0 F1 A5 9D C9 15 F0<69>
1BB0:D7 D0 D8 A9 39 2C A9 3A<E3>
1BB8:AE 81 04 C9 16 D0 05 A9<D6>
1BC0:3B 4C 84 E9 48 20 E5 EB<B5>
1BC8:A0 8C 20 3D EB 68 2C A9<C4>
1BD0:38 48 20 E5 EB A0 88 20<8D>
1BD8:3D EB 20 E5 EB A0 8A 20<C6>
1BE0:3D EB 20 E5 EB 68 4C 84<D7>
1BE8:E9 AD 81 04 C9 16 F0 36<02>
1BF0:A9 3D 2C A9 54 48 AD 81<B8>
1BF8:04 C9 1C D0 21 20 48 C1<B9>
1C00:20 A6 F0 68 AA A9 3F E0<B4>
1C08:84 F0 02 A9 37 20 A5 E9<75>
1C10:A9 A0 A0 00 20 A5 E9 8A<89>
1C18:4C 84 E9 A9 3C 48 20 7A<C4>
1C20:C8 20 4C E5 68 2C A9 55<76>
1C28:4C 84 E9 20 A6 F0 A5 92<0E>
1C30:99 00 DC A5 93 99 00 DD<66>
1C38:60 A9 3E 20 84 E9 20 A6<BC>
1C40:F0 AA A9 1F 4C A1 E9 20<AC>
1C48:48 C1 C9 00 D0 03 20 48<00>
1C50:C1 49 80 A8 20 48 C1 B9<80>
1C58:00 DB 85 98 98 60 20 A6<9D>
1C60:F0 A9 A2 4C A5 E9 AD 81<87>
1C68:04 C9 16 D0 05 A9 3F 4C<2D>
1C70:84 E9 A9 40 2C A9 41 2C<10>
1C78:A9 42 2C A9 43 2C A9 44<03>
1C80:2C A9 45 2C A9 46 2C A9<E4>
1C88:47 48 20 70 C8 20 4C E5<B1>
1C90:68 4C 84 E9 20 E5 EB A0<E9>
1C98:8C 20 3D EB A9 49 2C A9<40>
1CA0:48 48 20 84 EC 20 E5 EB<AF>
1CA8:A0 88 20 3D EB 20 E5 EB<F7>
1CB0:A0 8A 20 3D EB 4C E9 F0<2D>
1CB8:20 70 C8 C9 84 08 D0 02<08>
1CC0:C6 9B 20 4C E5 20 F1 C4<98>
1CC8:A5 9D C9 17 F0 0F C9 18<2A>
1CD0:F0 0E C9 62 F0 3A C9 6A<36>
1CD8:F0 33 4C 5F F5 A2 4C 2C<37>
1CE0:A2 4E 28 D0 01 E8 A9 27<10>
1CE8:A0 00 20 A5 E9 8A 20 84<00>
1CF0:E9 20 7A C8 A5 9B C9 03<9E>
1CF8:D0 37 AD 01 D8 AC 02 D8<F5>
1D00:20 A5 E9 EE 01 D9 A5 9D<11>
1D08:C9 16 D0 E5 60 A2 50 2C<C8>
1D10:A2 52 28 D0 01 E8 A9 2F<D9>
1D18:A0 00 20 A5 E9 8A 20 84<A8>
1D20:E9 20 A6 F0 20 A5 E9 EE<DF>
1D28:01 D9 A5 9D C9 16 D0 F1<A8>
1D30:60 20 00 E4 4F 4E 2E 47<B3>
1D38:4F 54 4F 2F 47 4F 53 55<FE>
1D40:42 20 65 78 70 00 4C 9D<F8>
1D48:C0 C6 9B 20 31 C9 4C 4C<3A>
1D50:E5 20 4C E5 C6 9A C6 9A<D7>
1D58:C6 9A A9 57 4C 84 E9 20<45>
1D60:EF E4 7D 9B 9B 9B 9B 20<FC>
1D68:20 54 75 72 62 6F 20 42<A8>
1D70:41 53 49 43 20 58 4C 20<96>
1D78:43 6F 6D 70 69 6C 65 72<04>
1D80:20 56 65 72 73 69 6F 6E<AC>
1D88:20 31 2E 31 9B 20 20 3D<AA>
1D90:3D 3D 3D 3D 3D 3D 3D 3D<AA>
1D98:3D 3D 3D 3D 3D 3D 3D 3D<92>
1DA0:3D 3D 3D 3D 3D 3D 3D 3D<9A>
1DA8:3D 3D 3D 3D 3D 3D 3D 3D<62>
1DB0:3D 3D 9B 9B 57 65 6C 63<D7>
1DB8:68 65 73 20 50 72 6F 67<CE>
1DC0:72 61 6D 6D 20 73 6F 6C<59>
1DC8:6C 20 63 6F 6D 70 69 6C<BB>
1DD0:69 65 72 74 20 77 65 72<64>
1DD8:64 65 6E 3F 9B 9B 20 47<09>
1DE0:65 62 65 6E 20 53 69 65<7D>
1DE8:20 7A 75 65 72 73 74 20<E7>
1DF0:64 69 65 20 4C 61 75 66<78>
1DF8:77 65 72 6B 73 6E 75 6D<B3>
1E00:6D 65 72 9B 9B 20 7A 77<C3>
1E08:69 73 63 68 65 6E 20 31<BA>
1E10:20 75 6E 64 20 38 20 65<34>
1E18:69 6E 2C 20 75 6E 64 20<F5>
1E20:77 0B 68 6C 65 6E 20 64<8F>
1E28:61 73 9B 9B 20 50 72 6F<10>
1E30:67 72 61 6D 6D 20 64 61<2C>
1E38:6E 6E 20 6D 69 74 20 64<12>
1E40:65 6E 20 50 66 65 69 6C<08>
```

Listing 1. Turbo-Basic-Compiler (Fortsetzung)

```
1E48:74 61 73 74 65 6E 9B 9B<1D>
1E50:20 1B 1E 2C 20 1B 1F 2C<C1>
1E58:20 1B 1C 20 75 6E 64 20<2F>
1E60:1B 1D 2E 20 44 61 6E 6E<27>
1E68:20 52 45 54 55 52 4E 20<9F>
1E70:64 72 0A 63 6B 65 6E 2E<DC>
1E78:9B 9B 9B 7F 43 6F 6E 74<68>
1E80:72 6F 6C 20 44 20 3D 20<4C>
1E88:52 0A 63 6B 6B 65 68 72<B3>
1E90:20 7A 75 6D 20 44 4F 53<D4>
1E98:9B 9B 7F 43 6F 6E 74 72<75>
1EA0:6F 6C 20 52 20 3D 20 52<E0>
1EA8:65 62 6F 6F 74 9B 00 20<AE>
1EB0:F7 C0 29 7F C9 04 F0 07<A2>
1EB8:C9 12 D0 06 4C B4 F4 4C<9D>
1EC0:D8 F4 C9 31 90 E9 C9 39<16>
1EC8:B0 E5 8D A3 20 48 20 1C<CF>
1ED0:C1 A9 24 8D 99 21 68 29<A5>
1ED8:1F 8D 9A 21 A9 1A 8D 9B<8D>
1EE0:21 A2 10 20 83 C4 A9 03<9A>
1EE8:9D 42 03 A9 A2 9D 44 03<47>
1EF0:A9 20 9D 45 03 A9 06 9D<E8>
1EF8:4A 03 A9 00 9D 4B 03 20<67>
1F00:88 C4 20 A6 F4 A9 00 85<5C>
1F08:92 A9 03 85 8E 20 27 C1<26>
1F10:C9 30 B0 24 A2 0D 20 27<9F>
1F18:C1 20 18 C2 CA D0 F7 20<6D>
1F20:27 C1 C9 9B D0 F9 E6 92<C5>
1F28:A6 92 E0 3F F0 0A C6 8E<4E>
1F30:D0 DB 20 E7 E4 4C 68 F3<A1>
1F38:A9 00 85 54 20 EA E4 A9<CA>
1F40:17 85 54 A2 10 20 83 C4<63>
1F48:A9 00 85 93 A5 92 D0 03<23>
1F50:4C BE F1 18 A5 58 69 78<95>
1F58:85 AB A5 59 69 00 85 AC<46>
1F60:A9 03 85 8E A6 93 F0 1C<21>
1F68:18 A5 AB 69 0D 85 AB 90<25>
1F70:02 E6 AC C6 8E D0 0A A9<8B>
1F78:03 85 8E E6 AB D0 02 E6<F0>
1F80:AC CA D0 E4 A0 0C B1 AB<8A>
1F88:09 80 91 AB 88 10 F7 20<DA>
1F90:F7 C0 A6 93 C9 9B F0 49<39>
1F98:C9 1C F0 27 C9 2D F0 23<70>
1FA0:C9 1D F0 34 C9 3D F0 30<16>
1FA8:C9 1E F0 19 C9 2B F0 15<4A>
1FB0:C9 1F F0 26 C9 2A F0 22<6C>
1FB8:C9 31 90 D3 C9 39 B0 CF<42>
1FC0:4C 29 F3 CA CA CA 30 C7<8A>
1FC8:86 93 A0 0C B1 AB 29 7F<E8>
1FD0:91 AB 88 10 F7 4C B2 F3<D0>
1FD8:E8 E8 E8 E4 92 B0 B0 90<8B>
1FE0:E7 A9 44 8D 80 04 AD A3<45>
1FE8:20 8D 81 04 A9 3A 8D 82<7A>
1FF0:04 A2 03 A0 01 B1 AB 29<3C>
1FF8:7F F0 07 18 69 20 9D 80<D7>
2000:04 E8 C8 C0 0C F0 0C C0<69>
2008:09 D0 EA A9 2E 9D 80 04<91>
2010:E8 D0 E2 A9 20 9D 80 04<C3>
2018:E8 E0 28 D0 F8 BD 7F 04<FB>
2020:38 E9 20 9D 98 21 CA D0<29>
2028:F4 A2 10 A9 03 9D 42 03<67>
2030:A9 80 9D 44 03 A9 04 9D<40>
2038:45 03 A9 04 9D 4A 03 A9<75>
2040:00 9D 4B 03 20 88 C4 A9<D1>
2048:7D 20 18 C2 20 EA E4 20<0E>
2050:EA E4 4C EA E4 20 EF E4<28>
2058:7F 7F D2 C5 C2 CF CF D4<8B>
2060:A0 CA AF CE A0 BF 00 20<4E>
2068:F7 C0 C9 4A F0 08 20 EF<90>
2070:E4 9C 00 4C 0E F3 4C 9C<BF>
2078:20 20 EF E4 7F 7F C4 CF<35>
2080:D3 A0 CA AF CE A0 BF 00<A4>
2088:20 F7 C0 C9 4A D0 DF 4C<B6>
2090:96 20 A9 6F 85 AB A9 22<13>
2098:85 AC A0 00 B1 AB 29 07<AB>
20A0:C9 07 F0 31 B1 AB A0 03<94>
20A8:C9 20 F0 14 29 0F C9 0D<BA>
20B0:F0 0E C9 0C F0 0A 88 C9<C5>
20B8:08 F0 04 C9 0A D0 01 88<1F>
20C0:18 98 65 AB 85 AB 90 02<83>
20C8:E6 AC C5 80 A5 AC E5 81<39>
20D0:90 C8 4C A3 F6 B1 AB C9<A0>
20D8:07 F0 4D C9 0F F0 27 C9<0B>
20E0:17 F0 20 C9 1F F0 3A C9<2E>
20E8:27 F0 0F C9 37 F0 0E C9<84>
20F0:3F F0 0A C9 2F D0 09 4C<B0>
20F8:A3 F5 4C A6 F5 4C 85 F6<92>
2100:4C 36 E5 A2 84 2C A2 82<4E>
2108:A9 A9 91 AB 18 C8 B1 AB<91>
2110:75 00 91 AB A0 03 B1 AB<88>
2118:75 01 91 AB A0 04 4C 1F<5E>
2120:F5 C8 20 2C F6 4C 91 F5<8E>
2128:C8 B1 AB AA C8 B1 AB 20<35>
2130:E6 F5 A0 02 91 AB 88 8A<A1>
2138:91 AB 88 A9 4C 91 AB A0<E3>
2140:03 4C 1F F5 A9 00 2C A9<98>
2148:FF 85 D2 A9 A2 91 AB C8<A8>
2150:B1 AB 85 D3 A0 05 24 D2<FF>
2158:30 06 20 2C F6 4C C8 F5<60>
2160:B1 AB AA C8 B1 AB 20 E6<4E>
2168:F5 A0 06 91 AB 88 8A 91<F4>
2170:AB 18 A5 AB 69 02 85 AB<5E>
2178:A5 AC 69 00 85 AC C6 D3<78>
2180:D0 D2 A0 05 4C 1F F5 85<B1>
2188:D5 86 D4 A5 86 85 D6 A5<02>
2190:87 85 D7 A0 00 B1 D6 C5<93>
2198:D4 C8 B1 D6 E5 D5 B0 09<4D>
21A0:C8 C8 C8 D0 F0 E6 D7 D0<95>
21A8:EC D0 07 88 B1 D6 C5 D4<78>
21B0:F0 12 20 4A F6 20 00 E4<30>
21B8:5A 65 69 6C 65 20 66 65<7E>
21C0:68 6C 74 00 C8 C8 B1 D6<34>
21C8:AA C8 B1 D6 60 B1 AB A8<89>
21D0:BE 00 DC B9 00 DD 10 AF<BD>
21D8:20 4A F6 20 00 E4 4C 61<06>
21E0:62 65 6C 20 66 65 68 6C<3D>
21E8:74 00 60 A5 AB A6 AC 85<97>
21F0:D4 86 D5 A5 86 85 D6 A5<39>
21F8:87 85 D7 A0 00 C8 C8 B1<53>
2200:D6 C5 D4 C8 B1 D6 E5 D5<F3>
2208:B0 07 C8 D0 F0 E6 D7 D0<90>
2210:EC 38 A5 D6 E9 06 85 D6<88>
2218:B0 02 C6 D7 B1 D6 85 93<57>
2220:88 B1 D6 85 92 60 A9 A9<5B>
2228:91 AB C8 B1 AB AA BD 00<DC>
2230:DC 91 AB A0 03 BD 00 DD<DE>
2238:91 AB 10 03 20 37 F6 A0<4B>
2240:04 4C 1F F5 20 F3 EC 10<81>
2248:03 4C 1F F7 48 A0 02 B1<EA>
2250:8A AA 88 B1 8A 20 4E F6<C7>
2258:68 F0 1D C9 01 F0 2D C9<F8>
2260:02 F0 3A C9 03 F0 48 20<FD>
2268:00 E4 4E 45 58 54 20 66<6C>
2270:65 68 6C 74 00 4C A3 F6<72>
2278:20 00 E4 45 58 49 54 20<44>
2280:6F 68 6E 65 20 2E 2E 2E<9A>
2288:00 4C A3 F6 20 00 E4 4C<A8>
2290:4F 4F 50 20 66 65 68 6C<1D>
2298:74 00 4C A3 F6 20 00 E4<4E>
22A0:55 4E 54 49 4C 20 66 65<C6>
22A8:68 6C 74 00 4C A3 F6 20<27>
22B0:00 E4 57 45 4E 44 20 66<52>
22B8:65 68 6C 74 00 4C A3 F6<4A>
22C0:A6 8D F0 23 A5 8C 20 4E<D0>
22C8:F6 20 00 E4 45 4E 44 49<7E>
22D0:46 20 66 65 68 6C 74 00<0E>
22D8:A0 02 B1 8C AA 88 B1 8C<28>
22E0:85 8C 86 8D 8A D0 DD 60<10>
22E8:20 BD F8 A9 00 85 A2 20<6E>
22F0:EF E4 9B 46 69 6C 65 6E<AC>
22F8:61 6D 65 20 66 75 65 72<F2>
2300:20 64 61 73 20 63 6F 6D<97>
2308:70 69 6C 69 65 72 74 65<B1>
2310:20 50 72 6F 67 72 61 6D<C8>
2318:6D 3F 9B 9B 00 A9 31 85<0B>
2320:A7 20 EF E4 9C 7F 7F 44<D7>
2328:00 A5 A7 20 18 C2 20 EF<65>
2330:E4 3A 2E 43 54 42 1E 1E<E2>
2338:1E 1E 00 A9 00 85 A5 A9<33>
2340:40 8D BE 02 A9 00 8D B6<BE>
2348:02 20 1E 21 C0 00 30 98<63>
2350:A4 A5 D0 0E C9 41 B0 2B<C0>
2358:C9 31 90 E3 C9 39 B0 DF<E9>
2360:90 BD C9 7E D0 0D 20 18<28>
2368:C2 A9 FE 20 18 C2 C6 A5<B8>
2370:4C 9E F7 C9 9B F0 25 C9<67>
2378:30 90 C4 C9 3A 90 04 C9<E1>
2380:41 90 BC C9 5B B0 B8 C0<20>
2388:08 B0 B4 99 83 04 48 A9<6C>
2390:FF 20 18 C2 68 20 18 C2<1B>
2398:E6 A5 D0 A3 A9 2E 99 83<21>
23A0:04 A9 43 99 84 04 A9 54<D6>
23A8:99 85 04 A9 42 99 86 04<E6>
23B0:A9 9B 99 87 04 A9 44 8D<96>
23B8:80 04 A5 A7 8D 81 04 A9<E6>
23C0:3A 8D 82 04 A9 80 8D 54<8D>
23C8:03 A9 04 8D 55 03 A9 08<53>
23D0:8D 5A 03 A9 00 8D 5B 03<5A>
23D8:A9 03 20 BF F8 A0 FB 20<F7>
23E0:E5 F8 A0 C2 20 E5 F8 A4<CF>
23E8:80 20 E5 F8 A4 81 20 E5<DE>
23F0:F8 A4 86 20 E5 F8 A4 87<A3>
23F8:20 E5 F8 A4 88 20 E5 F8<47>
2400:A4 89 20 E5 F8 A9 6F 8D<8C>
2408:54 03 A9 22 8D 55 03 38<6E>
2410:A5 88 E9 6F 8D 58 03 A5<09>
2418:89 E9 22 8D 59 03 A9 0B<0F>
2420:20 BF F8 20 BD F8 20 EF<26>
2428:E4 9B 20 20 20 20 20 4E<16>
2430:6F 63 68 20 65 69 6E 6D<65>
2438:61 6C 20 73 70 65 69 63<8A>
2440:68 65 72 6E 20 28 4A 2F<4C>
2448:4E 29 20 3F 00 20 1E 21<97>
2450:29 5F C9 4A F0 05 C9 4E<07>
2458:D0 F3 60 4C 47 F7 A9 0C<83>
2460:A2 10 8D 52 03 20 AF 20<BA>
2468:30 01 60 98 48 20 EF E4<1B>
2470:49 2F 4F 20 45 72 72 6F<58>
2478:72 20 2D 20 00 68 A0 00<4A>
2480:20 E8 C1 4C 47 F7 A2 10<11>
2488:20 FE 20 C0 00 30 DC 60<38>
2490:A9 04 A2 88 20 FF C3 A9<16>
2498:FF 91 AB C8 91 AB A5 80<6E>
24A0:C8 91 AB A5 81 C8 91 AB<CC>
24A8:A9 00 20 84 E9 A9 9B 20<C4>
24B0:A9 E9 20 F1 C4 A2 82 A9<EF>
24B8:03 20 FF C3 A9 FF 91 AB<A4>
24C0:C8 91 AB C8 A9 00 91 AB<7B>
24C8:60 00 DE FF DF ED E9 ED<75>
24D0:F9 0F FA BD 20 22 22 DD<0E>
24D8:21 AA EA C1 EA C1 FA 42<EC>
24E0:F0 EA EE E8 EE C8 EE B7<F8>
24E8:EC B4 EC 87 EF EF ED 2B<41>
24F0:EF EB ED 57 21 6B 21 C6<38>
24F8:ED BF ED DE ED D8 ED 86<08>
2500:ED 7D ED 54 C9 A9 FA 98<8B>
2508:FA A2 ED 66 FA 8B F9 84<6B>
2510:ED 7B ED C1 CA 4B EF 48<FE>
2518:EF CE C7 2D F7 32 F8 28<91>
2520:F8 ED F8 F4 ED 2A EE 06<E1>
2528:EE 1E EE FF ED 8C EE B1<C1>
2530:EE 6F F1 96 F1 0E F1 06<2C>
2538:F1 69 F1 96 F1 2E ED 4C<22>
2540:ED 47 ED B6 EA 3B F6 89<89>
2548:F0 BE FA 07 EF 10 EF 33<B2>
2550:EF 36 EF 30 EF 39 EF 95<43>
2558:FB C1 EF C3 EF 4A EE 4E<24>
2560:EE 78 EE 12 EE 50 FC 53<9A>
2568:FC 98 FC 9B FC 50 FC 53<37>
2570:FC 92 FC 95 FC DF FA 86<BB>
2578:F0 B1 EC AB EC E4 FA 00<02>
2580:00 00 00 00 00 00 00 00<CA>
2588:00 00 00 00 00 00 00 00<D2>
2590:00 00 00 00 00 00 00 00<DA>
2598:00 00 00 00 00 00 00 00<E2>
25A0:00 00 00 00 00 00 00 00<EA>
25A8:00 00 00 00 00 00 00 00<F2>
25B0:00 00 00 00 00 00 00 00<FA>
25B8:00 00 00 00 00 00 00 00<02>
25C0:00 00 00 00 00 00 00 00<0A>
25C8:00 00 00 00 00 C9 F1 C3<DA>
25D0:F1 BB F1 CF F1 B5 EB 3B<D7>
25D8:EC 77 EB 96 EB 83 EC 04<FF>
25E0:FE 55 FF 0A FE 3E FF D2<B7>
25E8:EA 6B FF 51 EB 74 FE 17<F2>
25F0:FE 86 FE 33 FE 92 FE 4A<0D>
25F8:EB 10 FE 2C FE 48 FE 4E<99>
2600:FE 06 FD 98 EC 10 FD 6A<E9>
2608:F2 76 F2 96 F2 80 F2 8A<7F>
2610:F2 A0 F2 3B F5 0E F2 FC<2F>
2618:F1 05 F2 17 F2 BD F2 B5<E9>
2620:F2 AA F2 F5 EA CE ED 13<D4>
2628:C0 3E EB D5 F2 DE F2 F5<7E>
2630:F2 E5 F2 EC F2 FC F2 02<11>
2638:FB 57 F2 54 F6 82 F4 AC<A4>
2640:F4 6F F6 D9 EB 55 FD 21<86>
2648:F4 4A F4 89 EC 1A EC 94<81>
2650:F3 09 E9 DD C3 0C F5 06<79>
2658:F5 31 21 00 F5 5E F4 75<09>
2660:F3 1E F5 12 F5 18 F5 24<83>
2668:F5 C7 F2 60 F2 C1 F4 D6<89>
2670:F3 DA F3 DE F3 E2 F3 40<2F>
2678:21 08 F6 14 F6 72 F6 34<EC>
2680:F4 20 F6 16 F4 A4 F3 A9<AC>
2688:F4 8E F4 25 FB 67 F2 73<0B>
2690:F2 93 F2 7D F2 87 F2 9D<EF>
2698:F2 38 F5 0B F2 F9 F1 02<92>
26A0:F2 14 F2 96 FF 9D FF A4<8E>
26A8:FF AF FF E5 F1 D5 F1 DB<F6>
26B0:F1 1B C7 34 F2 E7 F1 E1<4E>
26B8:F1 22 F2 03 EC 2B EC 46<DD>
26C0:F2 00 00 00 00 00 00 00<05>
26C8:00 00 00 00 00 00 00 00<14>
Laenge 9933 Bytes
```

Listing 1. Turbo-Basic-Compiler (Schluß)

Beginn: Listing 2 Turbo-Basic-Compiler-Runtime-Paket (Länge: 10876 Byte)

```
0000:FF FF 7D 20 C5 20 20 00<2E>
0008:00 A9 7D A0 20 85 0C 84<BF>
0010:0D A9 FE 8D 01 D3 20 D6<C2>
0018:EE 4C ED F9 6C E2 02 6C<C2>
0020:E0 02 EE 01 D3 20 93 20<BA>
0028:4C 22 21 EE 01 D3 20 96<03>
0030:20 4C 22 21 EE 01 D3 20<E2>
0038:56 E4 4C 22 21 20 AA EA<A6>
0040:EE 01 D3 6C 0A 00 20 AA<D3>
0048:EA EE 01 D3 4C 71 E4 00<EF>
0050:21 1C 21 EA 20 22 21 4C<BE>
0058:81 F0 44 3A 2A 2E 2A 9B<AA>
0060:53 3A 9B 50 3A 9B 45 3A<61>
0068:9B 00 EE 01 D3 6C D4 00<73>
0070:22 21 71 22 08 48 AD 01<55>
0078:D3 29 FC 09 02 8D 01 D3<57>
0080:68 28 60 EE 01 D3 85 D4<BB>
0088:84 D5 A0 00 B1 D4 CE 01<85>
0090:D3 60 EE 01 D3 85 D4 84<FC>
0098:D5 A0 01 B1 D4 A8 A2 00<DA>
00A0:A1 D4 CE 01 D3 60 4C 29<0D>
00A8:F5 C0 00 D0 F9 EE 01 D3<6C>
00B0:91 88 AD 01 D3 29 FC 09<41>
00B8:02 8D 01 D3 60 EE 01 D3<B9>
00C0:A2 00 81 88 98 A0 01 D0<3F>
00C8:E7 68 AA CE 01 D3 68 40<8E>
00D0:2C 0F D4 10 03 6C 00 02<F1>
00D8:48 8A 48 A9 21 48 A9 77<C1>
00E0:48 BA BD 05 01 48 D8 48<39>
00E8:8A 48 98 48 EE 01 D3 8D<DD>
00F0:0F D4 6C 22 02 48 A9 21<C2>
00F8:48 A9 79 48 08 EE 01 D3<47>
0100:6C 16 02 EE 01 D3 20 BB<2B>
0108:21 CE 01 D3 60 BD 47 03<E2>
0110:48 BD 46 03 48 BD 4A 03<25>
0118:85 2A BD 4B 03 85 2B 98<DA>
0120:A0 5C 60 EE 01 D3 06 2A<C1>
0128:20 00 00 CE 01 D3 60 85<B0>
0130:8C 84 8D EE 01 D3 A6 8D<F5>
0138:18 8A 65 89 85 89 18 8A<2E>
0140:65 8B 85 8B E8 A4 8C F0<AB>
0148:25 88 B1 88 91 8A 98 D0<49>
0150:F8 F0 1B C6 89 C6 8B 88<0A>
0158:B1 88 91 8A 88 B1 88 91<F0>
0160:8A 88 B1 88 91 8A 88 B1<28>
0168:88 91 8A 98 D0 E9 CA D0<92>
0170:E2 4C 22 21 85 8C 84 8D<B7>
0178:EE 01 D3 A0 00 A6 8D F0<A7>
0180:1D B1 88 91 8A C8 B1 88<CC>
0188:91 8A C8 B1 88 91 8A C8<5D>
0190:B1 88 91 8A C8 D0 EA E6<57>
0198:89 E6 8B CA D0 E3 A6 8C<A8>
01A0:F0 08 B1 88 91 8A C8 CA<47>
01A8:D0 F8 4C 22 21 EE 01 D3<D1>
01B0:20 E6 D8 CE 01 D3 60 EE<58>
01B8:01 D3 20 00 D8 CE 01 D3<BE>
01C0:60 20 ED E9 00 60 32 61<6A>
01C8:AE 24 E4 AC 25 E4 E8 D0<3A>
01D0:01 C8 8E D7 21 8C D8 21<90>
01D8:A9 00 8D 0E D4 78 A9 FE<73>
01E0:8D 01 D3 A9 7E 8D FA FF<6F>
01E8:A9 21 8D FB FF A9 A3 8D<19>
01F0:FE FF A9 21 8D FF FF A9<E8>
01F8:CC 20 88 60 A9 E0 20 88<8D>
0200:60 A9 40 8D 0E D4 58 A9<38>
0208:FF 8D 01 D3 20 AE 60 A9<5F>
0210:00 85 09 A5 0C A4 0D 8D<7A>
0218:7E 20 8C 7F 20 A9 7D A0<2B>
0220:20 85 0C 84 0D A9 FE 8D<B3>
0228:01 D3 A2 01 8E F8 03 86<9E>
0230:09 CA BE 44 02 A9 79 A2<7A>
0238:60 A0 0F 20 53 F4 4C 8B<C6>
0240:F9 44 31 3A 41 55 54 4F<F2>
0248:52 55 4E 2E 43 54 42 9B<BA>
0250:85 D5 A0 00 84 D4 A2 04<92>
0258:A9 FF 8D 01 D3 B1 D4 99<B0>
0260:00 05 C8 D0 F8 CE 01 D3<87>
0268:B9 00 05 91 D4 C8 D0 F8<C9>
0270:E6 D5 CA D0 E3 60 A2 10<BE>
0278:A9 D6 9D 44 03 A9 00 9D<95>
0280:45 03 9D 49 03 A9 04 9D<83>
0288:48 03 A9 07 9D 42 03 20<06>
0290:56 E4 30 5A A9 33 85 DA<4B>
0298:9D 44 03 A9 61 85 DB 9D<CF>
02A0:45 03 38 A5 D8 E5 D6 85<E0>
02A8:DC A5 D9 E5 D7 85 DD E6<BB>
02B0:DC D0 02 E6 DD A5 DC 9D<07>
02B8:48 03 A5 DD 9D 49 03 20<54>
02C0:56 E4 30 34 CE 01 D3 A0<DA>
02C8:00 A6 DD F0 0E B1 DA 91<8A>
02D0:D6 C8 D0 F9 E6 D7 E6 DB<6C>
02D8:CA D0 F2 A6 DC F0 08 B1<FE>
02E0:DA 91 D6 C8 CA D0 F8 A9<A3>
02E8:FF 8D 01 D3 30 88 A2 10<D1>
02F0:A9 0C 9D 42 03 4C 56 E4<EC>
02F8:6C 0A 00 E2 02 E3 02 00<F2>
0300:60 00 C0 B4 CB A9 05 85<F9>
0308:F4 A9 80 85 F3 60 A4 F2<0D>
0310:B1 F3 38 E9 30 C9 0A 60<59>
0318:84 F7 A8 20 09 E9 F8 98<88>
0320:F0 2E 4A 4A 4A 4A 85 F8<0C>
0328:98 29 07 A0 00 90 02 69<1A>
0330:07 46 F8 90 02 69 15 46<24>
0338:F8 90 02 69 31 46 F8 90<07>
0340:05 69 63 90 01 C8 46 F8<36>
0348:90 06 69 27 C8 90 01 C8<61>
0350:A6 F7 F0 6F 46 F7 90 07<10>
0358:69 55 C8 C8 90 01 C8 46<C3>
0360:F7 90 08 69 11 AA 98 69<26>
0368:05 A8 8A 46 F7 90 08 69<D3>
0370:23 AA 98 69 10 A8 8A 46<39>
0378:F7 90 08 69 47 AA 98 69<E0>
0380:20 A8 8A 46 F7 90 08 69<A5>
0388:95 AA 98 69 40 A8 8A 46<69>
0390:F7 90 0C 69 91 AA 98 69<11>
0398:81 A8 8A 90 02 E6 D5 46<2A>
03A0:F7 90 0E 69 83 AA 98 69<B1>
03A8:63 A8 A5 D5 69 01 85 D5<79>
03B0:8A 46 F7 90 0E 69 67 AA<B2>
03B8:98 69 27 A8 A5 D5 69 03<92>
03C0:85 D5 8A 84 D6 85 D7 A9<FF>
03C8:42 85 D4 4C EC C1 18 60<3A>
03D0:A5 E0 49 80 85 E0 A5 E0<17>
03D8:29 7F F0 F2 85 F7 A5 D4<57>
03E0:29 7F 38 E5 F7 B0 33 A5<D6>
03E8:D4 A4 E0 85 E0 84 D4 A5<83>
03F0:D5 A4 E1 85 E1 84 D5 A5<26>
03F8:D6 A4 E2 85 E2 84 D6 A5<C5>
0400:D7 A4 E3 85 E3 84 D7 A5<FD>
0408:D8 A4 E4 85 E4 84 D8 A5<C0>
0410:D9 A4 E5 85 E5 84 D9 4C<C9>
0418:D1 C0 A8 F0 4B 88 F0 36<3B>
0420:88 F0 22 88 F0 10 88 D0<C5>
0428:A5 A5 E1 85 E5 84 E4 84<D9>
0430:E3 84 E2 4C 61 C1 A5 E2<C4>
0438:85 E5 A5 E1 85 E4 84 E3<4E>
0440:84 E2 4C 61 C1 A5 E3 85<F9>
0448:E5 A5 E2 85 E4 A5 E1 85<B1>
0450:E3 84 E2 4C 61 C1 A5 E4<27>
0458:85 E5 A5 E3 85 E4 A5 E2<8F>
0460:85 E3 A5 E1 85 E2 84 E1<B9>
0468:F8 A5 D4 45 E0 30 3A 18<AC>
0470:A5 D9 65 E5 85 D9 A5 D8<A8>
0478:65 E4 85 D8 A5 D7 65 E3<90>
0480:85 D7 A5 D6 65 E2 85 D6<A4>
0488:A5 D5 65 E1 85 D5 90 16<E9>
0490:A5 D8 85 D9 A5 D7 85 D8<13>
0498:A5 D6 85 D7 A5 D5 85 D6<62>
04A0:A9 01 85 D5 E6 D4 4C EC<9E>
04A8:C1 38 A5 D9 E5 E5 85 D9<E2>
04B0:A5 D8 E5 E4 85 D8 A5 D7<A3>
04B8:E5 E3 85 D7 A5 D6 E5 E2<3A>
04C0:85 D6 A5 D5 E5 E1 85 D5<0B>
04C8:B0 27 A5 D4 49 80 85 D4<6B>
04D0:38 98 E5 D9 85 D9 98 E5<A2>
04D8:D8 85 D8 98 E5 D7 85 D7<05>
04E0:98 E5 D6 85 D6 98 E5 D5<A6>
04E8:85 D5 4C EC C1 A6 DA D0<D9>
04F0:02 A2 00 D8 A4 D4 F0 6D<A9>
04F8:A5 D5 D0 5C 88 A5 D6 D0<16>
0500:47 88 A5 D7 D0 30 88 A5<86>
0508:D8 D0 1B 88 A5 D9 D0 08<8E>
0510:88 8A F0 51 85 D5 D0 40<3E>
0518:85 D5 86 D6 A9 00 85 D9<57>
0520:85 D8 85 D7 F0 32 85 D5<27>
0528:A5 D9 85 D6 86 D7 A9 00<CB>
0530:85 D9 85 D8 F0 22 85 D5<27>
0538:A5 D8 85 D6 A5 D9 85 D7<09>
0540:86 D8 A9 00 85 D9 F0 10<BB>
0548:85 D5 A5 D7 85 D6 A5 D8<8C>
0550:85 D7 A5 D9 85 D8 86 D9<E0>
0558:84 D4 98 29 7F C9 71 B0<46>
0560:08 C9 0F B0 03 20 09 E9<21>
0568:18 60 86 FC 84 FD 85 EF<D0>
0570:20 C9 C2 20 3F EC C6 EF<D4>
0578:20 0B E4 B0 37 A5 FC 69<8A>
0580:06 85 FC 90 02 E6 FD 20<29>
0588:3F EC 20 D1 C0 B0 25 C6<1F>
0590:EF F0 21 AD E0 05 85 E0<0B>
0598:AD E1 05 85 E1 AD E2 05<2D>
05A0:85 E2 AD E3 05 85 E3 AD<78>
05A8:E4 05 85 E4 AD E5 05 85<5C>
05B0:E5 4C 73 C2 60 A5 D4 85<4E>
05B8:E0 A5 D5 85 E1 A5 D6 85<99>
05C0:E2 A5 D7 85 E3 A5 D8 85<0E>
05C8:E4 A5 D9 85 E5 60 A5 D4<59>
05D0:8D E0 05 A5 D5 8D E1 05<A3>
05D8:A5 D6 8D E2 05 A5 D7 8D<6F>
05E0:E3 05 A5 D8 8D E4 05 A5<66>
05E8:D9 8D E5 05 60 A5 D4 8D<51>
05F0:E6 05 A5 D5 8D E7 05 A5<C8>
05F8:D6 8D E8 05 A5 D7 8D E9<62>
0600:05 A5 D8 8D EA 05 A5 D9<51>
0608:8D EB 05 60 A5 E0 8D E6<E1>
0610:05 A5 E1 8D E7 05 A5 E2<31>
0618:8D E8 05 A5 E3 8D E9 05<FF>
0620:A5 E4 8D EA 05 A5 E5 8D<3D>
0628:EB 05 60 A2 00 F0 05 A2<7C>
0630:06 2C A2 0C BD E0 05 85<13>
0638:D4 BD E1 05 85 D5 BD E2<A4>
0640:05 85 D6 BD E3 05 85 D7<C1>
0648:BD E4 05 85 D8 BD E5 05<15>
0650:85 D9 60 A2 0C 2C A2 06<45>
0658:BD E0 05 85 E0 BD E1 05<9C>
0660:85 E1 BD E2 05 85 E2 BD<7B>
0668:E3 05 85 E3 BD E4 05 85<0A>
0670:E4 BD E5 05 85 E5 60 A2<AD>
0678:05 BD 2B C4 95 E0 CA 10<E9>
0680:F8 20 0B E4 B0 F0 A9 00<C6>
0688:85 F1 A5 D4 85 F0 29 7F<1A>
0690:85 D4 C9 40 90 1E D0 DE<DD>
0698:A5 D5 29 F0 4A 85 F1 4A<61>
06A0:4A 65 F1 85 F1 A5 D5 29<82>
06A8:0F 65 F1 85 F1 A9 00 85<7A>
06B0:D5 20 EC C1 A9 0A A2 EF<62>
06B8:A0 C3 20 65 C2 20 08 E4<22>
06C0:A5 F1 F0 0F 4A 18 65 D4<9B>
06C8:30 16 85 D4 46 F1 90 03<5A>
06D0:20 67 C7 06 F0 90 0A 20<D1>
06D8:B0 C2 20 DD C3 4C 6D E5<16>
06E0:38 60 A9 40 85 D4 A0 01<95>
06E8:84 D5 88 84 D6 84 D7 84<23>
06F0:D8 84 D9 60 3D 17 94 19<71>
06F8:00 00 3D 57 33 05 00 00<BE>
0700:3E 05 54 76 62 00 3E 32<11>
0708:19 62 27 00 3F 01 68 60<2C>
0710:30 36 3F 07 32 03 27 41<08>
0718:3F 25 43 34 56 75 3F 66<27>
0720:27 37 30 50 40 01 15 12<DC>
0728:92 55 3F 99 99 99 99 99<F7>
0730:3F 43 42 94 48 19 86 FC<DF>
0738:84 FD 20 C9 C2 20 3F EC<D0>
0740:20 D1 C0 20 E8 C2 20 26<39>
0748:C3 20 3F EC 20 CB C0 20<13>
0750:51 C3 4C 6D E5 38 60 A9<27>
0758:05 D0 02 A9 00 85 F0 A5<3E>
0760:D4 30 F2 F0 F0 0A 49 80<5B>
0768:85 F1 A9 40 85 D4 A5 D5<9F>
0770:29 F0 F0 05 E6 F1 20 6F<81>
0778:C7 A2 D2 A0 C4 20 31 C4<CA>
0780:20 E8 C2 20 08 E4 A9 0A<FF>
0788:A2 D8 A0 C4 20 65 C2 20<54>
0790:51 C3 20 0B E4 A9 3F 85<B6>
0798:E0 A9 50 85 E1 A9 00 85<41>
07A0:E2 85 E3 85 E4 85 E5 20<A4>
07A8:D1 C0 20 B0 C2 A5 F1 10<B4>
07B0:05 18 49 FF 69 01 A0 00<EE>
07B8:20 13 C0 A5 F1 29 80 05<20>
07C0:D4 85 D4 20 D1 C0 A6 F0<1B>
07C8:F0 0B BD 2B C4 95 E0 CA<D6>
07D0:10 F8 4C 6D E5 18 60 40<BD>
07D8:03 16 22 77 66 3F 49 15<7F>
07E0:57 11 08 BF 51 70 49 47<6F>
07E8:08 3F 39 20 57 61 95 BF<DC>
07F0:04 39 63 03 55 3F 10 09<3C>
07F8:30 12 64 3F 09 39 08 04<50>
0800:60 3F 12 42 58 47 42 3F<11>
0808:17 37 12 06 08 3F 28 95<36>
0810:29 71 17 3F 86 85 88 96<8A>
0818:44 3E 16 05 44 49 00 BE<EE>
0820:95 68 38 45 00 3F 02 68<89>
0828:79 94 16 BF 04 92 78 90<73>
0830:80 3F 07 03 15 20 00 BF<C9>
0838:08 92 29 12 44 3F 11 08<78>
0840:40 09 11 BF 14 28 31 56<BA>
0848:04 3F 19 99 98 77 44 BF<B4>
0850:33 33 33 31 13 3F 99 99<57>
```

Listing 2. Turbo-Basic-Compiler-Runtime-Paket. Geben Sie das Programm bitte mit AMPEL von Seite 87 ein.

```
0858:99 99 99 3F 78 53 98 16<28>
0860:34 38 60 A9 04 24 D4 10<06>
0868:06 A9 02 D0 02 A9 01 85<1C>
0870:F0 A5 D4 29 7F 85 D4 A6<A2>
0878:FB BD 14 C6 85 E0 BD 15<8A>
0880:C6 85 E1 BD 16 C6 85 E2<C1>
0888:BD 17 C6 85 E3 BD 18 C6<4E>
0890:85 E4 BD 19 C6 85 E5 20<AF>
0898:6D E5 B0 C5 A5 D4 29 7F<24>
08A0:38 E9 40 30 24 C9 04 10<5E>
08A8:B8 AA B5 D5 85 F1 29 10<75>
08B0:F0 02 A9 02 18 65 F1 29<CB>
08B8:03 65 F0 85 F0 86 F1 A9<1E>
08C0:00 95 D5 CA 10 FB 20 EC<14>
08C8:C1 46 F0 90 09 20 B0 C2<6D>
08D0:20 DD C3 20 CB C0 20 E8<02>
08D8:C2 20 08 E4 B0 83 A9 06<F9>
08E0:A2 F6 A0 C5 20 65 C2 20<03>
08E8:51 C3 20 0B E4 46 F0 90<2E>
08F0:09 18 A5 D4 F0 04 49 80<1A>
08F8:85 D4 60 BD 03 55 14 99<E2>
0900:39 3E 01 60 44 27 52 BE<75>
0908:46 81 75 43 55 3F 07 96<5B>
0910:92 62 39 BF 64 59 64 08<26>
0918:67 40 01 57 07 96 32 40<A6>
0920:90 00 00 00 00 3F 01 74<ED>
0928:53 29 25 A9 00 85 F0 85<E8>
0930:F1 A5 D4 29 7F C9 40 30<D0>
0938:15 A5 D4 29 80 85 F0 E6<27>
0940:F1 A9 7F 25 D4 85 D4 A2<53>
0948:50 A0 C5 20 31 C4 20 E8<85>
0950:C2 20 08 E4 B0 3B A9 0B<51>
0958:A2 14 A0 C5 20 65 C2 B0<2A>
0960:30 20 51 C3 20 0B E4 B0<0D>
0968:28 A5 F1 F0 13 A2 05 BD<7E>
0970:56 C5 95 E0 CA 10 F8 20<1F>
0978:D1 C0 A5 F0 05 D4 85 D4<6D>
0980:A5 FB F0 0D A2 05 BD 20<AF>
0988:C6 95 E0 CA 10 F8 20 6D<E9>
0990:E5 60 38 60 18 60 A9 00<CB>
0998:85 F1 A5 D4 30 F4 F0 F4<C0>
09A0:C9 3F F0 05 18 69 01 85<22>
09A8:F1 A9 06 85 EF A9 3F 85<83>
09B0:D4 20 E8 C2 20 B0 C2 20<41>
09B8:DD C3 E6 D5 20 CB C0 20<C1>
09C0:51 C3 20 0B E4 A5 D4 8D<CB>
09C8:EC 05 A5 D5 8D ED 05 A5<BB>
09D0:D6 8D EE 05 A5 D7 8D EF<CB>
09D8:05 A5 D8 8D F0 05 A5 D9<20>
09E0:8D F1 05 20 B0 C2 20 2A<0E>
09E8:C3 20 6D E5 20 4E C3 20<40>
09F0:CB C0 20 1B C7 A5 D4 F0<EF>
09F8:0C 20 4E C3 20 D1 C0 C6<89>
0A00:EF 10 C2 30 03 20 2D C3<80>
0A08:A5 F1 F0 0F 4A 18 65 D4<6B>
0A10:E9 1F 85 D4 46 F1 90 03<70>
0A18:20 67 C7 18 60 4C 09 E9<4B>
0A20:A5 D4 29 7F C9 0F 90 F5<57>
0A28:20 B0 C2 F8 A2 00 A0 04<88>
0A30:18 A5 D9 65 E5 85 D9 A5<BB>
0A38:D8 65 E4 85 D8 A5 D7 65<7E>
0A40:E3 85 D7 A5 D6 65 E2 85<E5>
0A48:D6 A5 D5 65 E1 85 D5 8A<E2>
0A50:69 00 AA 88 D0 DB D8 8A<6B>
0A58:F0 1A A5 D8 85 D9 A5 D7<C6>
0A60:85 D8 A5 D6 85 D7 A5 D5<4E>
0A68:85 D6 86 D5 A5 D5 C9 10<54>
0A70:90 31 E6 D4 A5 D5 C9 10<31>
0A78:90 27 4A 66 D6 66 D7 66<8A>
0A80:D8 66 D9 4A 66 D6 66 D7<08>
0A88:66 D8 66 D9 4A 66 D6 66<1B>
0A90:D7 66 D8 66 D9 4A 66 D6<9D>
0A98:66 D7 66 D8 66 D9 85 D5<3E>
0AA0:60 C6 D4 A9 00 06 D9 26<5B>
0AA8:D8 26 D7 26 D6 26 D5 2A<8A>
0AB0:06 D9 26 D8 26 D7 26 D6<0F>
0AB8:26 D5 2A 06 D9 26 D8 26<F1>
0AC0:D7 26 D6 26 D5 2A 06 D9<1B>
0AC8:26 D8 26 D7 26 D6 26 D5<C1>
0AD0:2A 60 60 A5 8B D0 FB A5<62>
0AD8:D6 05 D7 F0 F5 A0 00 84<9D>
0AE0:DB 84 DC B1 D4 10 02 C6<92>
0AE8:DB 20 54 C8 0A 0A 85 8C<86>
0AF0:A9 00 2A 06 8C 2A 6D F4<50>
0AF8:02 85 8D 20 9E C8 B0 D2<49>
0B00:A0 08 84 DD A4 DC B1 8C<40>
0B08:45 DB 85 DA A6 ED A4 E3<F1>
0B10:06 DA B1 DE 3D C0 05 90<16>
0B18:03 1D C8 05 91 DE C6 DD<9E>
0B20:F0 07 20 8D CA C4 E1 90<7D>
0B28:E7 20 62 CA E6 DC A5 DC<46>
0B30:C9 08 B0 06 65 8A C5 E0<BF>
0B38:90 C6 A5 88 69 07 85 88<AB>
0B40:90 02 E6 89 E6 D4 D0 02<04>
0B48:E6 D5 A5 D6 D0 02 C6 D7<89>
0B50:C6 D6 4C D2 C7 40 20 60<58>
0B58:00 A8 0A 0A 2A 2A 29 03<85>
0B60:AA 98 5D 50 C8 60 18 18<18>
0B68:0C 18 30 30 60 60 C0 C0<D7>
0B70:C0 C0 18 0C C0 C0 28 14<FF>
0B78:14 0A 0A 14 14 28 28 28<57>
0B80:28 28 28 28 14 28 00 00<7D>
0B88:00 02 03 02 03 02 03 01<C6>
0B90:01 01 00 00 03 02 00 01<88>
0B98:03 07 00 F0 FC FE 04 02<D2>
0BA0:01 38 60 A5 57 29 0F AA<92>
0BA8:A9 00 85 DF A5 8A 0A 26<1E>
0BB0:DF 0A 26 DF 65 8A 90 02<DF>
0BB8:E6 DF 0A 85 DE 26 DF BD<45>
0BC0:61 C8 85 E0 C5 8A F0 D9<CE>
0BC8:90 D7 BC 81 C8 84 E2 B9<B6>
0BD0:95 C8 8D C0 05 BD 71 C8<98>
0BD8:85 E1 4A 4A 4A 4A AA F0<0E>
0BE0:07 06 DE 26 DF CA D0 F9<D2>
0BE8:18 A5 DE 65 58 85 DE A5<27>
0BF0:DF 65 59 85 DF A5 89 85<A4>
0BF8:E4 A5 88 85 E3 A4 E2 84<B0>
0C00:ED F0 0C 39 91 C8 85 ED<3D>
0C08:46 E4 66 E3 88 D0 F9 A5<69>
0C10:E4 D0 8E A5 E3 C5 E1 B0<C6>
0C18:88 A6 E2 D0 0A A5 C8 20<A9>
0C20:54 C8 8D C8 05 18 60 BC<67>
0C28:91 C8 84 E4 BD 98 C8 85<CA>
0C30:EE A5 C8 0D C0 05 4D C0<9E>
0C38:05 99 C8 05 A6 EE 0A CA<4D>
0C40:D0 FC 88 10 F4 A4 E4 AD<18>
0C48:C0 05 99 C0 05 A6 EE 38<87>
0C50:2A CA D0 FB 88 10 F3 18<1C>
0C58:60 85 8A 98 D0 FA 20 9E<60>
0C60:C8 B0 F5 A5 0E 85 8C A5<73>
0C68:0F 85 8D AD E5 02 E9 06<EC>
0C70:85 E7 AD E6 02 E9 00 85<51>
0C78:E8 18 A5 8C 69 03 85 8C<DC>
0C80:90 02 E6 8D C5 E7 A5 8D<9F>
0C88:E5 E8 90 03 4C 61 FA A6<C9>
0C90:ED A4 E3 20 A8 CA F0 03<AD>
0C98:4C 3C CA 20 9D CA 20 96<1B>
0CA0:CA 98 30 0B 20 A8 CA D0<3D>
0CA8:06 20 9D CA 4C 99 C9 20<92>
0CB0:8D CA 98 A0 00 91 8C 8A<83>
0CB8:0A 0A 0A C8 91 8C A4 E3<F3>
0CC0:A6 ED 20 8D CA C4 E1 B0<82>
0CC8:0B 20 A8 CA D0 06 20 9D<7B>
0CD0:CA 4C BD C9 20 96 CA 98<A8>
0CD8:A0 02 91 8C 8A 88 11 8C<58>
0CE0:91 8C A4 8A C8 C4 E0 B0<56>
0CE8:23 20 60 CA 20 7C CA A0<44>
0CF0:01 B1 8C 29 07 C5 ED C8<E6>
0CF8:B1 8C E5 E3 90 0B A0 00<7E>
0D00:B1 8C 09 80 91 8C 4C 74<BC>
0D08:C9 20 6E CA A4 8A 88 C4<19>
0D10:E0 B0 20 20 6E CA 20 7C<62>
0D18:CA A0 01 B1 8C 29 07 C5<46>
0D20:ED C8 B1 8C E5 E3 90 08<6D>
0D28:A0 00 B1 8C 29 7F 10 D4<18>
0D30:20 60 CA A0 01 B1 8C 29<B2>
0D38:07 AA C8 B1 8C A8 20 8D<30>
0D40:CA 20 8D CA 86 ED 84 E3<1F>
0D48:38 A5 8C E9 03 85 8C B0<B9>
0D50:02 C6 8D C5 0E D0 06 A5<1B>
0D58:8D C5 0F F0 15 A0 00 B1<A2>
0D60:8C 10 B6 30 8A E6 8A 18<2F>
0D68:A5 DE 65 E1 85 DE 90 02<44>
0D70:E6 DF 60 C6 8A 38 A5 DE<62>
0D78:E5 E1 85 DE B0 02 C6 DF<DB>
0D80:60 A0 00 B1 8C 29 7F 85<91>
0D88:E3 C8 B1 8C 4A 4A 4A 85<86>
0D90:ED 60 E4 E4 E8 90 03 A2<19>
0D98:00 C8 60 CA 10 03 A6 E4<19>
0DA0:88 60 B1 DE 3D C0 05 1D<DE>
0DA8:C8 05 91 DE 60 B1 DE 1D<8E>
0DB0:C0 05 5D C0 05 F0 04 AD<30>
0DB8:C8 05 60 AD C8 05 F0 03<F3>
0DC0:A9 00 60 A9 01 60 20 E8<77>
0DC8:F7 AD FE 02 48 8E FE 02<DF>
0DD0:A9 7D 20 F4 ED 68 8D FE<A3>
0DD8:02 60 C9 FF 90 02 C0 50<28>
0DE0:8A 69 00 85 D4 60 98 29<9C>
0DE8:F0 4A 85 D4 4A 4A 65 D4<D5>
0DF0:85 D4 98 29 0F A6 D6 E0<02>
0DF8:50 65 D4 85 D4 60 A2 00<7B>
0E00:A4 D5 A5 D4 B6 D5 38 E9<6E>
0E08:40 90 CF F0 D9 C9 02 F0<23>
0E10:45 B0 42 A5 D6 29 F0 4A<FA>
0E18:85 D4 4A 4A 65 D4 85 D4<33>
0E20:A5 D6 29 0F A6 D7 E0 50<BA>
0E28:65 D4 85 D4 98 29 0F AA<0B>
0E30:BD A1 CB 65 D4 85 D4 BD<8B>
0E38:AB CB 65 D5 85 D5 98 29<05>
0E40:F0 F0 12 4A 4A 4A AA A5<A4>
0E48:D4 7D 8D CB 85 D4 A5 D5<A1>
0E50:7D 8E CB 85 D5 60 C0 07<6E>
0E58:B0 FB A5 D7 29 F0 4A 85<4F>
0E60:D4 4A 4A 65 D4 85 D4 A5<7D>
0E68:D7 29 0F A6 D8 E0 50 65<3A>
0E70:D4 85 D4 B9 81 CB 65 D4<FA>
0E78:85 D4 B9 88 CB 65 D5 85<06>
0E80:D5 A4 D6 D0 A7 60 00 10<AE>
0E88:20 30 40 50 60 00 27 4E<24>
0E90:75 9C C3 EA E8 03 D0 07<37>
0E98:B8 0B A0 0F 88 13 70 17<C0>
0EA0:58 1B 40 1F 28 23 00 64<DB>
0EA8:C8 2C 90 F4 58 BC 20 84<5A>
0EB0:00 00 00 01 01 01 02 02<EE>
0EB8:03 03 00 E4 BD FF 18 4C<B3>
0EC0:09 E9 18 60 38 60 20 B0<C6>
0EC8:C2 A5 D4 F0 F5 A5 E0 F0<96>
0ED0:ED 45 D4 29 80 85 EE A5<4C>
0ED8:E0 29 7F 85 E0 A5 D4 29<C9>
0EE0:7F 38 E9 40 38 65 E0 30<72>
0EE8:DB 05 EE A8 20 0B E8 85<FD>
0EF0:DA 85 DB 85 DC 85 DD 85<81>
0EF8:DE 85 DF 84 D4 A0 07 46<01>
0F00:E5 90 30 18 A5 DE 79 EE<B9>
0F08:20 85 DE A5 DD 79 E6 20<26>
0F10:85 DD A5 DC 79 DE 20 85<A1>
0F18:DC A5 DB 79 D6 20 85 DB<4B>
0F20:A5 DA 79 CE 20 85 DA A5<84>
0F28:D9 79 C6 20 85 D9 88 10<3A>
0F30:CE 30 05 F0 03 88 10 C7<C1>
0F38:A0 07 46 E4 90 30 18 A5<0D>
0F40:DD 79 EE 20 85 DD A5 DC<B1>
0F48:79 E6 20 85 DC A5 DB 79<00>
0F50:DE 20 85 DB A5 DA 79 D6<19>
0F58:20 85 DA A5 D9 79 CE 20<61>
0F60:85 D9 A5 D8 79 C6 20 85<F1>
0F68:D8 88 10 CE 30 05 F0 03<2C>
0F70:88 10 C7 A0 07 46 E3 90<50>
0F78:30 18 A5 DC 79 EE 20 85<47>
0F80:DC A5 DB 79 E6 20 85 DB<7E>
0F88:A5 DA 79 DE 20 85 DA A5<6A>
0F90:D9 79 D6 20 85 D9 A5 D8<DF>
0F98:79 CE 20 85 D8 A5 D7 79<AB>
0FA0:C6 20 85 D7 88 10 CE 30<11>
0FA8:05 F0 03 88 10 C7 A0 07<10>
0FB0:46 E2 90 30 18 A5 DB 79<FD>
0FB8:EE 20 85 DB A5 DA 79 E6<79>
0FC0:20 85 DA A5 D9 79 DE 20<1F>
0FC8:85 D9 A5 D8 79 D6 20 85<59>
0FD0:D8 A5 D7 79 CE 20 85 D7<55>
0FD8:A5 D6 79 C6 20 85 D6 88<B1>
0FE0:10 CE 30 05 F0 03 88 10<21>
0FE8:C7 A0 07 46 E1 90 30 18<D4>
0FF0:A5 DA 79 EE 20 85 DA A5<B1>
0FF8:D9 79 E6 20 85 D9 A5 D8<68>
1000:79 DE 20 85 D8 A5 D7 79<20>
1008:D6 20 85 D7 A5 D6 79 CE<5C>
1010:20 85 D6 A5 D5 79 C6 20<FE>
1018:85 D5 88 10 CE 30 05 F0<4B>
1020:03 88 10 C7 4C E8 C1 18<EF>
1028:60 38 60 A5 E0 F0 FA A5<04>
1030:D4 F0 F4 45 E0 29 80 85<32>
1038:EE A5 E0 29 7F 85 E0 A5<B2>
1040:D4 29 7F 38 E5 E0 18 69<27>
1048:40 30 DE 05 EE A8 20 0B<71>
1050:E8 85 E6 85 E7 85 E8 85<3C>
1058:E9 85 EA 85 E0 85 DA 84<DC>
1060:D4 A0 00 A5 E0 D9 C6 20<3F>
1068:D0 28 A5 E1 D9 CE 20 D0<AF>
1070:21 A5 E2 D9 D6 20 D0 1A<FC>
1078:A5 E3 D9 DE 20 D0 13 A5<48>
1080:E4 D9 E6 20 D0 0C A5 E5<66>
1088:D9 EE 20 D0 05 A2 00 4C<8C>
1090:02 E8 90 2A A5 E5 F9 EE<F1>
1098:20 85 E5 A5 E4 F9 E6 20<01>
10A0:85 E4 A5 E3 F9 DE 20 85<13>
10A8:E3 A5 E2 F9 D6 20 85 E2<6B>
10B0:A5 E1 F9 CE 20 85 E1 A5<CD>
10B8:E0 F9 C6 20 85 E0 26 D5<9E>
10C0:C8 C0 08 D0 9E A0 00 A5<0B>
10C8:E1 D9 C6 20 D0 28 A5 E2<94>
10D0:D9 CE 20 D0 21 A5 E3 D9<0C>
10D8:D6 20 D0 1A A5 E4 D9 DE<CD>
```

Listing 2. Turbo-Basic-Compiler-Runtime-Paket (Fortsetzung)

```
10E0:20 D0 13 A5 E5 D9 E6 20<2F>
10E8:D0 0C A5 E6 D9 EE 20 D0<F5>
10F0:05 A2 01 4C 02 E8 90 2A<0C>
10F8:A5 E6 F9 EE 20 85 E6 A5<E6>
1100:E5 F9 E6 20 85 E5 A5 E4<6E>
1108:F9 DE 20 85 E4 A5 E3 F9<CC>
1110:D6 20 85 E3 A5 E2 F9 CE<70>
1118:20 85 E2 A5 E1 F9 C6 20<C8>
1120:85 E1 26 D6 C8 C0 08 D0<2C>
1128:9E A0 00 A5 E2 D9 C6 20<79>
1130:D0 28 A5 E3 D9 CE 20 D0<F5>
1138:21 A5 E4 D9 D6 20 D0 1A<16>
1140:A5 E5 D9 DE 20 D0 13 A5<F2>
1148:E6 D9 E6 20 D0 0C A5 E7<31>
1150:D9 EE 20 D0 05 A2 02 4C<5A>
1158:02 E8 90 2A A5 E7 F9 EE<B2>
1160:20 85 E7 A5 E6 F9 E6 20<2A>
1168:85 E6 A5 E5 F9 DE 20 85<CE>
1170:E5 A5 E4 F9 D6 20 85 E4<96>
1178:A5 E3 F9 CE 20 85 E3 A5<9F>
1180:E2 F9 C6 20 85 E2 26 D7<5B>
1188:C8 C0 08 D0 9E A0 00 A5<CD>
1190:E3 D9 C6 20 D0 28 A5 E4<5F>
1198:D9 CE 20 D0 21 A5 E5 D9<DE>
11A0:D6 20 D0 1A A5 E6 D9 DE<97>
11A8:20 D0 13 A5 E7 D9 E6 20<13>
11B0:D0 0C A5 E8 D9 EE 20 D0<67>
11B8:05 A2 03 4C 02 E8 90 2A<E5>
11C0:A5 E8 F9 EE 20 85 E8 A5<B5>
11C8:E7 F9 E6 20 85 E7 A5 E6<52>
11D0:F9 DE 20 85 E6 A5 E5 F9<B7>
11D8:D6 20 85 E5 A5 E4 F9 CE<61>
11E0:20 85 E4 A5 E3 F9 C6 20<E0>
11E8:85 E3 26 D8 C8 C0 08 D0<E5>
11F0:9E A0 00 A5 E4 D9 C6 20<C3>
11F8:D0 28 A5 E5 D9 CE 20 D0<0D>
1200:21 A5 E6 D9 D6 20 D0 1A<18>
1208:A5 E7 D9 DE 20 D0 13 A5<BB>
1210:E8 D9 E6 20 D0 0C A5 E9<FE>
1218:D9 EE 20 D0 05 A2 04 4C<2B>
1220:02 E8 90 2A A5 E9 F9 EE<BC>
1228:20 85 E9 A5 E8 F9 E6 20<34>
1230:85 E8 A5 E7 F9 DE 20 85<98>
1238:E7 A5 E6 F9 D6 20 85 E6<A5>
1240:A5 E5 F9 CE 20 85 E5 A5<71>
1248:E4 F9 C6 20 85 E4 26 D9<30>
1250:C8 C0 08 D0 9E A5 D5 D0<3F>
1258:64 A0 00 A5 E5 D9 C6 20<9E>
1260:D0 28 A5 E6 D9 CE 20 D0<FA>
1268:21 A5 E7 D9 D6 20 D0 1A<A1>
1270:A5 E8 D9 DE 20 D0 13 A5<85>
1278:E9 D9 E6 20 D0 0C A5 EA<E9>
1280:D9 EE 20 D0 05 A2 05 4C<A2>
1288:02 E8 90 2A A5 EA F9 EE<09>
1290:20 85 EA A5 E9 F9 E6 20<D4>
1298:85 E9 A5 E8 F9 DE 20 85<D1>
12A0:E8 A5 E7 F9 D6 20 85 E7<8D>
12A8:A5 E6 F9 CE 20 85 E6 A5<1C>
12B0:E5 F9 C6 20 85 E5 26 DA<36>
12B8:C8 C0 08 D0 9E 4C E8 C1<A9>
12C0:36 D5 C8 C0 08 D0 F9 F0<8B>
12C8:F4 F8 18 A5 E5 8D F5 20<99>
12D0:65 E5 8D F4 20 A5 E4 8D<39>
12D8:ED 20 65 E4 8D EC 20 A5<3F>
12E0:E3 8D E5 20 65 E3 8D E4<65>
12E8:20 A5 E2 8D DD 20 65 E2<6C>
12F0:8D DC 20 A5 E1 8D D5 20<2B>
12F8:65 E1 8D D4 20 A9 00 8D<26>
1300:CD 20 69 00 8D CC 20 A2<1F>
1308:02 BD F2 20 7D F2 20 9D<27>
1310:F1 20 BD EA 20 7D EA 20<D7>
1318:9D E9 20 BD E2 20 7D E2<03>
1320:20 9D E1 20 BD DA 20 7D<E6>
1328:DA 20 9D D9 20 BD D2 20<FD>
1330:7D D2 20 9D D1 20 BD CA<21>
1338:20 7D CA 20 9D C9 20 CA<E3>
1340:D0 C7 AD F4 20 6D F2 20<AF>
1348:8D F1 20 AD EC 20 6D EA<72>
1350:20 8D E9 20 AD E4 20 6D<32>
1358:E2 20 8D E1 20 AD DC 20<9E>
1360:6D DA 20 8D D9 20 AD D4<FD>
1368:20 6D D2 20 8D D1 20 AD<C9>
1370:CC 20 6D CA 20 8D C9 20<01>
1378:A2 02 BD EF 20 7D EF 20<E7>
1380:9D EE 20 BD E7 20 7D E7<5E>
1388:20 9D E6 20 BD DF 20 7D<C3>
1390:DF 20 9D DE 20 BD D7 20<6D>
1398:7D D7 20 9D D6 20 BD CF<7E>
13A0:20 7D CF 20 9D CE 20 BD<33>
13A8:C7 20 7D C7 20 9D C6 20<55>
13B0:CA 10 C7 A5 D5 85 E1 A5<06>
13B8:D6 85 E2 A5 D7 85 E3 A5<E9>
13C0:D8 85 E4 A5 D9 85 E5 A9<5A>
13C8:00 85 D4 85 D5 85 D6 85<C0>
13D0:D7 85 D8 85 D9 60 68 85<84>
13D8:FC 68 85 FD E6 FC D0 02<F9>
13E0:E6 FD A2 00 A1 FC F0 07<C8>
13E8:A8 20 B1 21 4C 1E E9 A5<E4>
13F0:FD 48 A5 FC 48 60 C0 00<5E>
13F8:10 FB 98 85 C3 D8 A0 00<FC>
1400:84 93 8C FE 02 C9 80 D0<66>
1408:02 84 11 A2 FD 9A 68 85<02>
1410:D2 68 85 D3 48 A5 D2 48<B9>
1418:A5 82 85 D6 A5 83 85 D7<3A>
1420:A5 9B D0 3F A0 00 C8 C8<A5>
1428:B1 D6 C5 D2 C8 B1 D6 E5<65>
1430:D3 B0 07 C8 D0 F0 E6 D7<2C>
1438:D0 EC 38 A5 D6 E9 06 85<77>
1440:D6 B0 02 C6 D7 B1 D6 85<C6>
1448:BB 88 B1 D6 85 BA A5 BC<7D>
1450:A4 BD 30 0F A2 80 86 BD<7F>
1458:E4 C3 D0 04 A6 95 F0 03<60>
1460:4C 42 F0 20 18 E9 FD 9B<38>
1468:46 65 68 6C 65 72 20 2D<EE>
1470:20 00 A5 C3 A0 00 20 B1<7F>
1478:EC 20 18 E9 20 69 6E 20<B3>
1480:5A 65 69 6C 65 20 00 A5<63>
1488:BA A4 BB 20 B1 EC 20 18<95>
1490:E9 20 28 24 00 38 A5 D2<78>
1498:E9 02 48 A5 D3 E9 00 20<6B>
14A0:94 EA 68 20 94 EA A9 29<DC>
14A8:20 F4 ED A9 00 85 93 20<5F>
14B0:18 E9 9B 50 72 6F 67 72<FD>
14B8:61 6D 6D 65 6E 64 65 20<4A>
14C0:3A 20 C4 6F 73 2C 20 D2<53>
14C8:75 6E 20 6F 64 65 72 20<2F>
14D0:CC 6F 61 64 20 3F 00 A9<B4>
14D8:40 8D BE 02 A9 00 8D B6<51>
14E0:02 20 D1 21 C9 44 F0 0B<7B>
14E8:C9 4C F0 0A C9 52 D0 E7<7C>
14F0:4C 80 20 4C B4 20 20 18<3D>
14F8:E9 9B 46 69 6C 65 6E 61<EF>
1500:6D 65 20 44 3A 00 20 4A<CE>
1508:F1 A0 FF C8 B1 F3 C9 9B<7E>
1510:F0 29 C9 3A D0 F5 C8 B1<D4>
1518:F3 C9 9B D0 F9 A9 2E 91<A6>
1520:F3 C8 A9 43 91 F3 C8 A9<C8>
1528:54 91 F3 C8 A9 42 91 F3<FE>
1530:C8 A5 F3 A6 F4 20 53 F4<DE>
1538:4C 8B F9 C8 B9 7F 05 99<ED>
1540:81 05 88 D0 F7 A9 44 8D<69>
1548:80 05 A9 3A 8D 81 05 4C<EC>
1550:4B EA 48 4A 4A 4A 4A 20<CF>
1558:9F EA 68 29 0F 09 30 C9<32>
1560:3A 90 02 69 06 4C F4 ED<88>
1568:A2 70 20 C4 EA 8A 38 E9<F5>
1570:10 AA D0 F6 A2 08 A9 00<E4>
1578:9D 00 D2 CA 10 FA 60 20<91>
1580:EA EE A9 0C 9D 42 03 20<0C>
1588:AB 20 30 01 60 4C 3C E9<B3>
1590:A6 92 E6 92 A5 D4 9D 00<BF>
1598:DE A5 D5 9D 2A DE A5 D6<31>
15A0:9D 54 DE A5 D7 9D 7E DE<36>
15A8:A5 D8 9D A8 DE A5 D9 9D<FD>
15B0:D2 DE 60 20 B0 C2 C6 92<13>
15B8:A6 92 BD 00 DE 85 D4 BD<1B>
15C0:2A DE 85 D5 BD 54 DE 85<C9>
15C8:D6 BD 7E DE 85 D7 BD A8<65>
15D0:DE 85 D8 BD D2 DE 85 D9<BD>
15D8:60 C6 92 A6 92 BD 00 DE<B9>
15E0:85 E0 BD 2A DE 85 E1 BD<A0>
15E8:54 DE 85 E2 BD 7E DE 85<84>
15F0:E3 BD A8 DE 85 E4 BD D2<CC>
15F8:DE 85 E5 60 A6 92 E6 92<D6>
1600:9D 00 DE 98 9D 2A DE 60<E1>
1608:86 D3 20 13 C0 A6 D3 A5<48>
1610:D4 9D 00 D8 A5 D5 9D 00<84>
1618:D9 A5 D6 9D 00 DA A5 D7<F6>
1620:9D 00 DB A5 D8 9D 00 DC<55>
1628:A5 D9 9D 00 DD A5 11 F0<02>
1630:01 60 4C 54 FE BD 00 D8<6E>
1638:85 D4 BD 00 D9 85 D5 BD<74>
1640:00 DA 85 D6 BD 00 DB 85<19>
1648:D7 BD 00 DC 85 D8 BD 00<CB>
1650:DD 85 D9 60 BD 00 D8 85<47>
1658:E0 BD 00 D9 85 E1 BD 00<DA>
1660:DA 85 E2 BD 00 DB 85 E3<7C>
1668:BD 00 DC 85 E4 BD 00 DD<26>
1670:85 E5 60 85 FC 84 FD A0<68>
1678:00 B1 FC 85 D4 C8 B1 FC<CD>
1680:85 D5 C8 B1 FC 85 D6 C8<32>
1688:B1 FC 85 D7 C8 B1 FC 85<AC>
1690:D8 C8 B1 FC 85 D9 60 85<45>
1698:FC 84 FD A0 00 B1 FC 9D<E5>
16A0:00 D8 C8 B1 FC 9D 00 D9<12>
16A8:C8 B1 FC 9D 00 DA C8 B1<E6>
16B0:FC 9D 00 DB C8 B1 FC 9D<55>
16B8:00 DC C8 B1 FC 9D 00 DD<1F>
16C0:60 9D 00 D9 A9 40 9D 00<D8>
16C8:D8 A9 00 9D 00 DA 9D 00<77>
16D0:DB 9D 00 DC 9D 00 DD 60<5F>
16D8:85 D5 A9 40 85 D4 A9 00<1A>
16E0:85 D6 85 D7 85 D8 85 D9<CF>
16E8:60 20 1A EC 86 D3 20 96<9A>
16F0:EB 20 FC F1 A6 D3 4C 51<92>
16F8:EB 85 FC 84 FD A0 00 B1<CA>
1700:FC 85 E0 C8 B1 FC 85 E1<25>
1708:C8 B1 FC 85 E2 C8 B1 FC<AB>
1710:85 E3 C8 B1 FC 85 E4 C8<13>
1718:B1 FC 85 E5 60 85 FC 84<0D>
1720:FD A0 00 A5 E0 91 FC C8<50>
1728:A5 E1 91 FC C8 A5 E2 91<FC>
1730:FC C8 A5 E3 91 FC C8 A5<C2>
1738:E4 91 FC C8 A5 E5 91 FC<96>
1740:60 20 77 EB 4C CE ED 86<BD>
1748:D3 48 20 77 EB 68 20 F9<9F>
1750:F1 A6 D3 4C 51 EB 85 D4<AA>
1758:84 D5 A0 00 B1 D4 85 D6<EE>
1760:84 D7 E6 D4 D0 02 E6 D5<AF>
1768:60 20 98 EC 4C B7 EC 20<B5>
1770:13 C0 20 21 F4 A6 93 A5<30>
1778:D4 9D 44 03 A5 D5 9D 45<8F>
1780:03 A5 D6 9D 48 03 18 65<C9>
1788:94 85 94 A5 D7 9D 49 03<8F>
1790:05 D6 F0 05 A9 0B 4C C6<15>
1798:EA 60 4C 61 FA 85 99 84<BA>
17A0:9A 18 65 90 A8 A5 91 65<A1>
17A8:9A CD E6 02 90 09 D0 EA<DF>
17B0:CC E5 02 90 02 D0 E3 85<11>
17B8:0F 84 0E 38 A5 90 E5 8E<53>
17C0:85 8C A5 91 E5 8F 85 8D<AD>
17C8:18 A5 8E 85 97 85 88 65<42>
17D0:99 85 8A 85 8E A5 8F 85<E7>
17D8:98 85 89 65 9A 85 8B 85<9E>
17E0:8F A5 0E 85 90 A5 0F 85<F3>
17E8:91 4C E1 21 29 0F 06 8A<2B>
17F0:06 8A 06 8A 06 8A 05 8A<DB>
17F8:A6 88 E0 05 90 03 4C 29<F1>
1800:F5 9D C4 02 60 A0 78 06<E4>
1808:8C 2C A0 00 8C 08 D2 A2<1F>
1810:03 8E 0F D2 06 8C A6 8C<9E>
1818:E0 08 B0 E2 85 8C A5 88<83>
1820:9D 00 D2 98 F0 07 E8 EB<A5>
1828:A5 89 9D 00 D2 A5 8A 0A<A6>
1830:0A 0A 0A 05 8C 9D 01 D2<24>
1838:60 38 24 18 20 BF ED 4C<4E>
1840:8A ED 38 24 18 20 C6 ED<73>
1848:A5 C8 8D FB 02 A2 60 A9<3E>
1850:0C 9D 4A 03 A9 00 9D 4B<32>
1858:03 A9 11 69 00 4C C6 EA<D5>
1860:48 A9 06 20 C1 EA 68 A8<BE>
1868:29 F0 49 1C AA A9 0D 86<A4>
1870:88 84 8A A2 21 A0 02 20<DC>
1878:53 F4 4C 4A EE 85 55 84<31>
1880:56 86 54 60 85 54 98 F0<82>
1888:0C 4C 29 F5 20 F9 CA A5<23>
1890:D4 A4 D5 B0 F4 60 20 BF<49>
1898:ED 4C E1 ED 20 C6 ED A2<90>
18A0:60 A4 C8 20 B1 21 4C 38<7F>
18A8:E9 A9 00 85 93 A9 9B 2C<A4>
18B0:A9 20 A6 93 A8 E6 94 20<49>
18B8:B1 21 4C 38 E9 20 C6 ED<D8>
18C0:A9 60 85 93 86 88 20 69<0A>
18C8:EF A0 00 A6 88 4C 4A EB<A3>
18D0:86 88 20 81 EF A0 00 A6<11>
18D8:88 4C 4A EB 86 88 A0 07<C7>
18E0:20 2C EE A6 88 4C 51 EB<42>
18E8:A0 0B A9 06 A6 93 9D 48<A3>
18F0:03 A9 D4 9D 44 03 A9 00<91>
18F8:9D 45 03 9D 49 03 98 4C<05>
1900:C6 EA 86 93 85 88 84 8A<58>
1908:A9 03 85 8C A6 93 A5 8C<0F>
1910:9D 42 03 A5 88 9D 4A 03<26>
1918:A5 8A 9D 4B 03 A5 D4 9D<78>
1920:44 03 A5 D5 9D 45 03 20<33>
1928:A1 EF 20 AB 20 20 BA EF<F9>
1930:BC 43 03 4C 38 E9 86 88<2E>
1938:A9 0D A6 93 9D 42 03 20<E3>
1940:AB 20 98 A0 00 A6 88 4C<C2>
1948:4A EB 86 88 84 8A A9 26<C4>
1950:A6 93 20 C6 EA BD 4C 03<F8>
1958:BC 4D 03 A6 88 20 4A EB<31>
1960:A6 93 BD 4E 03 A0 00 A6<43>
```

Listing 2. Turbo-Basic-Compiler-Runtime-Paket (Fortsetzung)

```
1968:8A 4C 4A EB 4C 29 F5 A6<81>
1970:93 9D 4E 03 98 D0 F5 A5<18>
1978:88 9D 4C 03 A5 89 9D 4D<A1>
1980:03 A9 25 4C C6 EA A9 07<D5>
1988:20 C1 EA A2 08 A0 00 A9<FA>
1990:10 4C B1 ED A9 00 20 C1<09>
1998:EA A9 C0 85 6A A2 0C A0<CC>
19A0:00 A9 13 4C B1 ED A9 00<DE>
19A8:C9 08 B0 14 0A 0A 0A 0A<EE>
19B0:85 93 AA A9 00 85 94 85<6B>
19B8:F2 A4 C9 D0 02 E6 C9 60<7C>
19C0:A9 14 4C 3D E9 A9 07 A2<47>
19C8:21 A0 06 20 53 F4 A2 70<0D>
19D0:A9 06 A0 00 20 44 EE 20<5A>
19D8:6F EF C0 88 F0 0B 20 38<60>
19E0:E9 A2 00 20 F6 ED 4C 19<36>
19E8:EF A2 70 4C C4 EA A9 20<24>
19F0:2C A9 23 2C A9 24 2C A9<68>
19F8:21 A2 70 86 93 A0 00 84<38>
1A00:88 84 8A 4C 4C EE A2 0B<AF>
1A08:2C A2 07 48 8A A6 93 9D<01>
1A10:42 03 A5 88 9D 44 03 A5<F4>
1A18:89 9D 45 03 68 9D 48 03<F9>
1A20:98 9D 49 03 4C C9 EA 20<1B>
1A28:6F EF 4C 38 E9 A6 93 A9<0A>
1A30:00 9D 48 03 9D 49 03 A9<7D>
1A38:07 9D 42 03 4C AB 20 20<F4>
1A40:D1 21 4C 38 E9 A5 C9 85<16>
1A48:88 A4 94 C8 C4 88 90 05<E2>
1A50:18 65 C9 90 F2 20 F2 ED<EA>
1A58:A5 94 C5 88 D0 F7 60 A4<32>
1A60:D6 A5 D7 F0 02 A0 FF 84<13>
1A68:88 B1 D4 85 8A A5 D4 85<EF>
1A70:8C A5 D5 85 8D A9 9B 2C<5F>
1A78:A5 8A A4 88 91 8C 60 18<88>
1A80:24 38 6E 16 21 A9 3F A0<B0>
1A88:21 8D E0 02 8C E1 02 A2<52>
1A90:10 A9 04 A0 00 20 44 EE<A8>
1A98:20 69 EF C9 FF D0 5C 20<B1>
1AA0:69 EF C9 FF D0 55 A9 3F<52>
1AA8:A0 21 8D E2 02 8C E3 02<AE>
1AB0:20 69 EF 9D 44 03 20 69<86>
1AB8:EF 9D 45 03 3D 44 03 C9<2E>
1AC0:FF F0 E3 20 69 EF 48 20<7C>
1AC8:69 EF A8 68 38 FD 44 03<A9>
1AD0:9D 48 03 98 FD 45 03 9D<6E>
1AD8:49 03 FE 48 03 D0 03 FE<11>
1AE0:49 03 20 C9 EA 20 99 20<2B>
1AE8:A2 10 DE 43 03 F0 B7 20<B6>
1AF0:C4 EA 2C 16 21 10 03 4C<FA>
1AF8:A2 20 60 A9 01 4C 3D E9<84>
1B00:20 4A F0 68 68 6C D8 00<FB>
1B08:85 D4 84 D5 A5 82 85 D6<83>
1B10:A5 83 85 D7 A0 00 B1 D6<AD>
1B18:C5 D4 C8 B1 D6 E5 D5 B0<67>
1B20:09 C8 C8 C8 D0 F0 E6 D7<23>
1B28:D0 EC D0 13 88 B1 D6 C5<00>
1B30:D4 D0 0C C8 C8 B1 D6 85<77>
1B38:D8 C8 B1 D6 85 D9 60 A9<BF>
1B40:09 4C 3D E9 A9 00 A8 85<1D>
1B48:D4 84 D5 98 30 29 A5 80<5C>
1B50:A4 81 84 87 85 86 A0 00<95>
1B58:B1 86 C5 D4 C8 B1 86 E5<A3>
1B60:D5 B0 0B C8 B1 86 65 86<FB>
1B68:90 EA E6 87 B0 E6 A5 86<8F>
1B70:D0 02 C6 87 C6 86 60 4C<89>
1B78:29 F5 A9 06 4C 3D E9 86<BD>
1B80:88 A0 00 B1 86 C9 9B F0<72>
1B88:08 E6 86 D0 15 E6 87 D0<49>
1B90:11 A0 03 B1 86 F0 E3 18<B4>
1B98:A5 86 69 04 85 86 90 02<51>
1BA0:E6 87 A0 00 B1 86 C9 2C<1C>
1BA8:F0 07 C9 9B F0 03 C8 D0<56>
1BB0:F3 A5 86 A6 87 20 53 F4<02>
1BB8:18 A5 86 65 D6 85 86 90<A9>
1BC0:02 E6 87 60 20 C1 F0 A6<D2>
1BC8:88 4C 74 FE 20 C1 F0 A5<AD>
1BD0:D4 A4 D5 A2 00 85 F3 84<9A>
1BD8:F4 86 F2 A5 D6 48 20 65<A7>
1BE0:22 B0 0A 68 C5 F2 D0 05<6D>
1BE8:A6 88 4C 51 EB A9 08 4C<3B>
1BF0:3D E9 86 88 A5 93 D0 0A<8D>
1BF8:A5 F2 D0 2A A9 3F 20 F4<54>
1C00:ED 2C 86 88 A5 F2 D0 1E<91>
1C08:20 00 C0 A6 93 A5 F3 9D<54>
1C10:44 03 A5 F4 9D 45 03 A9<C8>
1C18:00 9D 48 03 A9 01 9D 49<A6>
1C20:03 A9 05 20 C6 EA 60 20<5D>
1C28:44 F1 4C 72 F1 20 34 F1<70>
1C30:20 65 22 B0 B8 A4 F2 B1<BA>
1C38:F3 C9 9B F0 0B C9 2C D0<10>
1C40:AC E6 F2 A6 88 4C 51 EB<61>
1C48:A9 00 85 F2 F0 F5 20 44<DC>
1C50:F1 4C 99 F1 20 34 F1 18<FF>
1C58:A5 F3 65 F2 85 F3 90 02<00>
1C60:E6 F4 A0 00 B1 F3 C9 9B<1A>
1C68:F0 03 C8 D0 F7 A5 F3 A6<35>
1C70:F4 20 53 F4 A6 88 4C 74<AE>
1C78:FE A5 D4 F0 04 49 80 85<1B>
1C80:D4 20 1B EB 4C FC F1 20<10>
1C88:1B EB 4C 0E F2 20 F5 EA<57>
1C90:4C 17 F2 20 B0 C2 4C FC<0C>
1C98:F1 20 B0 C2 4C 0E F2 A0<81>
1CA0:C0 D0 04 A9 01 A0 40 85<43>
1CA8:E1 84 E0 A9 00 85 E2 85<57>
1CB0:E3 85 E4 85 E5 F0 03 20<95>
1CB8:3B EC 20 D1 C0 B0 1C 60<73>
1CC0:20 3B EC 20 CB C0 B0 13<A0>
1CC8:60 20 3B EC 20 0B E4 B0<03>
1CD0:0A 60 20 3B EC 20 6D E5<7C>
1CD8:B0 01 60 A9 0B 4C 3D E9<38>
1CE0:85 E1 A9 40 85 E0 A9 00<25>
1CE8:85 E2 85 E3 85 E4 85 E5<C3>
1CF0:F0 E3 85 E1 A9 40 85 E0<A6>
1CF8:A9 00 85 E2 85 E3 85 E4<F8>
1D00:85 E5 F0 C8 85 E1 A9 40<F3>
1D08:85 E0 A9 00 85 E2 85 E3<71>
1D10:85 E4 85 E5 60 A5 D4 F0<B3>
1D18:04 49 80 85 D4 60 A5 D4<4C>
1D20:29 7F 85 D4 60 20 3B EC<60>
1D28:20 03 F3 90 55 F0 53 B0<5B>
1D30:4E 20 3B EC 20 03 F3 F0<0A>
1D38:46 D0 47 20 3B EC 20 03<C0>
1D40:F3 90 3F B0 3A 20 3B EC<6F>
1D48:20 03 F3 90 32 F0 30 B0<1C>
1D50:31 20 3B EC 20 03 F3 90<D4>
1D58:26 B0 27 20 3B EC 20 03<45>
1D60:F3 F0 1F D0 1A 20 3B EC<A4>
1D68:A5 E0 25 D4 F0 11 D0 12<D4>
1D70:20 3B EC A5 E0 05 D4 F0<BD>
1D78:06 D0 07 A5 D4 F0 03 4C<85>
1D80:09 E9 4C DD C3 A5 D4 F0<F9>
1D88:F6 10 F7 20 C4 F2 A9 C0<5B>
1D90:85 D4 60 20 3A F3 90 EA<A6>
1D98:F0 E8 B0 E3 20 3A F3 F0<C8>
1DA0:DE D0 DF 20 3A F3 90 DA<72>
1DA8:B0 D5 20 3A F3 90 D0 F0<FD>
1DB0:CE B0 CF 20 3A F3 90 C7<5D>
1DB8:B0 C8 20 3A F3 F0 C3 D0<35>
1DC0:BE A5 D4 45 E0 D0 25 A5<0F>
1DC8:D5 C5 E1 D0 18 A5 D6 C5<FB>
1DD0:E2 D0 12 A5 D7 C5 E3 D0<6B>
1DD8:0C A5 D8 C5 E4 D0 06 A5<C0>
1DE0:D9 C5 E5 F0 06 6A 45 D4<C7>
1DE8:0A A9 01 60 30 05 A5 D4<22>
1DF0:C5 E0 60 06 E0 A9 01 60<FC>
1DF8:A0 00 A5 D6 D0 06 A5 D7<CC>
1E00:F0 05 C6 D7 C6 D6 AA 08<7C>
1E08:A5 E2 D0 06 A5 E3 F0 20<22>
1E10:C6 E3 C6 E2 28 F0 16 B1<A8>
1E18:D4 D1 E0 D0 0E E6 D4 D0<EB>
1E20:02 E6 D5 E6 E0 D0 D3 E6<AB>
1E28:E1 D0 CF B0 02 C8 18 60<6D>
1E30:28 38 60 38 AD E5 02 E5<7F>
1E38:0E AA AD E6 02 E5 0F A8<63>
1E40:8A 60 20 A1 EF A5 D4 A4<BD>
1E48:D5 85 F3 84 F4 A0 00 84<C9>
1E50:F2 60 20 84 F3 20 65 22<CC>
1E58:90 34 20 D0 F3 A9 12 4C<95>
1E60:3D E9 20 84 F3 84 D4 84<87>
1E68:D5 B1 F3 38 E9 30 90 1E<A2>
1E70:C9 17 B0 1A C9 0A 90 06<03>
1E78:E9 07 C9 0A 90 10 A2 04<A8>
1E80:06 D4 26 D5 CA D0 F9 05<4B>
1E88:D4 85 D4 C8 D0 DB 20 BA<AA>
1E90:EF 4C 00 C0 A2 00 F0 0A<4A>
1E98:A2 08 D0 06 A2 0C D0 02<E8>
1EA0:A2 14 18 86 D4 65 D4 AA<A5>
1EA8:BD 70 02 A0 00 60 4C 29<DC>
1EB0:F5 A0 00 A5 D5 F0 03 20<26>
1EB8:FE F3 A5 D4 48 4A 4A 4A<CA>
1EC0:4A 20 09 F4 68 29 0F 09<17>
1EC8:30 C9 3A 90 02 69 06 99<F5>
1ED0:80 05 C8 60 85 D4 84 D5<DE>
1ED8:20 F3 F3 A9 80 D0 30 20<AD>
1EE0:5B 22 A0 FF C8 B1 F3 10<31>
1EE8:FB 29 7F 91 F3 C8 A5 F3<DB>
1EF0:D0 1D AD FC 02 A0 00 C9<0C>
1EF8:C0 B0 12 A2 0E DD 2C F6<9E>
1F00:F0 0B CA 10 F8 20 81 EF<63>
1F08:8D C0 05 A0 01 A9 C0 A2<25>
1F10:05 86 D5 85 D4 84 D6 A9<58>
1F18:00 85 D7 60 A9 3F 85 D4<C1>
1F20:A2 05 AD 0A D2 29 F0 C9<82>
1F28:A0 B0 F7 85 DA AD 0A D2<C4>
1F30:29 0F C9 0A B0 F7 05 DA<55>
1F38:95 D4 CA D0 E5 4C EC C1<19>
1F40:20 B0 C2 20 5E F4 20 0E<85>
1F48:F2 4C C1 F4 A5 D4 29 7F<77>
1F50:38 E9 40 90 11 AA A9 00<66>
1F58:E0 05 90 02 A2 04 95 D5<1D>
1F60:CA 10 FB 4C EC C1 60 20<EE>
1F68:17 F2 A5 D4 29 7F 38 E9<1F>
1F70:40 90 44 AA A9 00 E0 04<1F>
1F78:B0 3A 95 D6 E8 90 F7 A5<21>
1F80:D4 29 7F 38 E9 40 90 2D<DC>
1F88:AA E0 04 B0 D9 A9 00 A8<30>
1F90:15 D6 94 D6 E8 E0 04 D0<45>
1F98:F7 24 D4 10 17 A8 F0 14<22>
1FA0:A9 C0 85 E0 A0 01 84 E1<C4>
1FA8:88 84 E2 84 E3 84 E4 84<78>
1FB0:E5 4C D1 C0 60 06 D4 20<6D>
1FB8:09 E9 90 F8 B0 E2 20 5E<CA>
1FC0:C5 B0 24 60 20 68 C5 B0<AD>
1FC8:1E 60 20 26 C6 B0 18 60<10>
1FD0:20 52 C4 B0 12 60 20 56<CA>
1FD8:C4 B0 0C 60 20 72 C3 B0<EC>
1FE0:06 60 20 91 C6 90 FA A9<53>
1FE8:03 4C 3D E9 A5 E0 30 F7<D4>
1FF0:4C 09 E9 4C DD C3 20 3B<ED>
1FF8:EC A5 E0 F0 F6 A5 D4 F0<2E>
2000:EB 10 26 29 7F 85 D4 A5<5F>
2008:E0 29 7F 38 E9 40 90 D7<F4>
2010:A2 04 C9 04 F0 0E B0 11<C3>
2018:AA A8 B9 E2 00 D0 C8 C8<E7>
2020:C0 04 D0 F6 B5 E1 4A B0<2A>
2028:01 18 A9 00 6A 48 A5 E0<9F>
2030:C9 40 D0 4D A5 E2 05 E3<5D>
2038:05 E4 05 E5 D0 43 A5 E1<9A>
2040:29 F0 4A 85 FC 4A 4A 65<A3>
2048:FC 85 FC A5 E1 29 0F 65<E0>
2050:FC 85 FC 20 C9 C2 46 FC<3F>
2058:B0 03 20 DD C3 20 E8 C2<AD>
2060:A5 FC F0 50 20 26 C3 20<B4>
2068:08 E4 B0 0F 46 FC 90 F7<E6>
2070:20 C9 C2 20 51 C3 20 0B<4E>
2078:E4 90 E2 4C 1D F2 4C 29<F9>
2080:F5 A5 E5 48 A5 E4 48 A5<6F>
2088:E3 48 A5 E2 48 A5 E1 48<A9>
2090:A5 E0 48 20 56 C4 B0 E6<09>
2098:68 85 E0 68 85 E1 68 85<65>
20A0:E2 68 85 E3 68 85 E4 68<51>
20A8:85 E5 20 0B E4 B0 CC 20<B2>
20B0:81 C3 B0 C7 68 10 04 05<F4>
20B8:D4 85 D4 60 85 E0 84 E1<22>
20C0:20 F8 EA 4C CE ED 20 FE<68>
20C8:F5 25 E0 AA 98 25 E1 A8<25>
20D0:8A 60 20 FE F5 05 E0 AA<76>
20D8:98 05 E1 A8 8A 60 20 FE<58>
20E0:F5 45 E0 AA 98 45 E1 A8<C2>
20E8:8A 60 9A 98 9D 9B B3 B5<83>
20F0:B0 B2 A6 3C 7C BC 27 67<AF>
20F8:A7 85 D4 A5 14 A6 11 F0<FB>
2100:10 C5 14 F0 F8 A5 D4 D0<0B>
2108:03 88 30 05 C6 D4 4C 3D<95>
2110:F6 60 20 C9 C2 20 07 C3<55>
2118:20 A9 F4 20 B0 C2 20 2A<03>
2120:C3 20 0E F2 20 B0 C2 20<11>
2128:26 C3 4C 05 F2 A2 5F 2C<10>
2130:A2 FF 86 DF 85 DA 84 DB<37>
2138:20 1B EB 20 F8 EA 18 A5<64>
2140:D4 65 DA 85 D4 A5 D5 65<DD>
2148:DB 85 D5 B0 22 38 A5 D6<48>
2150:E5 DA 85 D6 A5 D7 E5 DB<D8>
2158:85 D7 90 13 05 D6 F0 0F<A4>
2160:38 A5 D6 E5 E2 85 DC A5<74>
2168:D7 E5 E3 85 DD B0 04 A9<4A>
2170:00 A8 60 E6 DA D0 02 E6<DE>
2178:DB A0 00 B1 E0 51 D4 25<3C>
2180:DF D0 30 A5 D4 85 D8 A5<7F>
2188:D5 85 D9 A5 E0 85 E4 A5<49>
2190:E1 85 E5 A5 E3 85 DE A6<26>
2198:E2 E8 CA D0 04 C6 DE 30<43>
21A0:25 B1 D8 51 E4 25 DF D0<03>
21A8:0A C8 D0 EE E6 D9 E6 E5<1E>
21B0:4C DC F6 E6 D4 D0 02 E6<74>
21B8:D5 A5 DC D0 04 C6 DD 30<29>
21C0:AE C6 DC 4C B5 F6 A5 DA<AD>
21C8:A4 DB 60 85 D6 85 D7 A0<77>
21D0:08 16 E6 36 E5 26 D6 90<B7>
21D8:0B 18 B5 E6 65 D7 95 E6<50>
21E0:90 02 F6 E5 88 D0 EA 60<AB>
21E8:4C 29 F5 85 D4 98 05 8D<19>
```

Listing 2. Turbo-Basic-Compiler-Runtime-Paket (Fortsetzung)

```
21F0:D0 F6 A5 8C D0 02 A9 01<BD>
21F8:85 D5 A2 16 A9 00 95 D6<0C>
2200:CA 10 FB A5 D5 85 E0 E8<3B>
2208:20 0D F7 A5 D4 A2 02 20<19>
2210:0D F7 A5 D5 85 D6 A0 08<45>
2218:06 DC 26 DB 26 DA 06 D6<43>
2220:90 11 18 A5 DC 65 E8 85<97>
2228:DC A5 DB 65 E7 85 DB 90<D8>
2230:02 E6 DA 88 D0 E2 20 DA<C2>
2238:F7 20 FF F7 20 DA F7 20<C1>
2240:0D F8 20 F1 F7 20 FF F7<45>
2248:20 F1 F7 20 0D F8 24 E2<2C>
2250:30 21 E6 E1 18 A5 EB 65<59>
2258:E6 85 EB A5 EA 65 E5 85<D0>
2260:EA 90 02 E6 E9 38 A2 02<C7>
2268:B5 E2 F5 E9 95 E2 CA 10<BC>
2270:F7 30 C3 A5 E0 F0 2F C6<D0>
2278:E0 38 A5 DC E5 E8 85 DC<BC>
2280:A5 DB E5 E7 85 DB B0 02<D3>
2288:C6 DA 18 A2 02 B5 E2 75<60>
2290:DA 95 E2 CA 10 F7 30 9E<5A>
2298:18 A5 88 65 E0 85 55 A5<BE>
22A0:89 69 00 85 56 60 A2 00<13>
22A8:86 54 86 55 86 56 60 38<2C>
22B0:A5 88 E5 E0 85 55 A5 89<4F>
22B8:E9 00 85 56 60 18 A5 8A<96>
22C0:65 E1 85 54 A5 8B 69 00<F6>
22C8:F0 0E 60 38 A5 8A E5 E1<11>
22D0:85 54 A5 8B E9 00 D0 F2<12>
22D8:A4 C8 A2 60 20 B1 21 C0<80>
22E0:8D F0 E7 4C 38 E9 86 8A<82>
22E8:20 D2 EA 20 DD C3 A6 8A<B7>
22F0:86 8A A6 90 86 88 A6 91<AE>
22F8:86 89 20 66 FA B0 0C D0<82>
2300:0A A0 0C B1 90 C5 8A D0<7E>
2308:F1 F0 08 A5 88 85 90 A5<32>
2310:89 85 91 A9 10 20 3A FA<06>
2318:C6 92 A6 92 A0 00 BD 00<4A>
2320:DE 91 88 C8 BD 2A DE 91<02>
2328:88 C8 BD 54 DE 91 88 C8<1F>
2330:BD 7E DE 91 88 C8 BD A8<CE>
2338:DE 91 88 C8 BD D2 DE 91<8A>
2340:88 C8 A5 D4 91 88 C8 A5<B6>
2348:D5 91 88 C8 A5 D6 91 88<34>
2350:C8 A5 D7 91 88 C8 A5 D8<B0>
2358:91 88 C8 A5 D9 91 88 C8<7C>
2360:A5 8A 91 88 C8 A9 00 91<DC>
2368:88 C8 68 AA 18 69 05 91<9E>
2370:88 C8 68 48 69 00 91 88<6A>
2378:8A 48 A5 96 D0 26 A5 90<75>
2380:48 A5 91 48 A5 88 85 90<8E>
2388:A5 89 85 91 A5 D4 48 A6<B6>
2390:8A 20 77 EB 68 20 4C F9<96>
2398:90 07 68 85 91 68 85 90<8C>
23A0:60 68 68 60 38 60 A9 0D<55>
23A8:4C 3D E9 86 8A 20 66 FA<05>
23B0:B0 F4 D0 F2 A0 0C B1 90<05>
23B8:C5 8A D0 F1 A0 06 B1 90<F5>
23C0:48 85 E0 C8 B1 90 85 E1<75>
23C8:C8 B1 90 85 E2 C8 B1 90<10>
23D0:85 E3 C8 B1 90 85 E4 C8<B5>
23D8:B1 90 85 E5 A6 8A 20 77<CC>
23E0:EB 20 FC F1 A6 8A 20 51<1E>
23E8:EB 68 20 4C F9 90 1A 68<08>
23F0:68 A0 0F B1 90 48 88 B1<AA>
23F8:90 48 A9 10 18 65 90 85<26>
2400:90 85 0E 90 04 E6 91 E6<FF>
2408:0F 60 85 EC A0 00 B1 90<3F>
2410:C5 D4 D0 2A C8 B1 90 C5<31>
2418:D5 D0 1C C8 B1 90 C5 D6<16>
2420:D0 15 C8 B1 90 C5 D7 D0<AE>
2428:0E C8 B1 90 C5 D8 D0 07<CE>
2430:C8 B1 90 C5 D9 F0 06 6A<E3>
2438:45 EC 45 D4 0A 60 05 D4<93>
2440:45 EC 10 F9 6A 49 80 0A<2F>
2448:60 20 AA EA A2 70 A9 04<E0>
2450:A0 00 20 44 EE 20 69 EF<D2>
2458:C9 FB D0 6C 20 69 EF C9<E2>
2460:C2 D0 65 85 BD 85 9B A9<6E>
2468:20 8D 6F 22 A9 ED 8D 70<B6>
2470:22 A9 E9 8D 71 22 20 69<E7>
2478:EF 85 80 20 69 EF 85 81<DE>
2480:20 69 EF 85 82 20 69 EF<BD>
2488:85 83 20 69 EF 85 84 20<41>
2490:69 EF 85 85 A9 6F A0 22<2A>
2498:85 88 84 89 38 A5 84 E9<74>
24A0:6F AA A5 85 E9 22 A8 8A<30>
24A8:20 4B EF A2 FF 9A 86 11<D5>
24B0:86 BD E8 86 96 86 95 86<BE>
24B8:FB 86 9B 20 0F FA 20 AA<FF>
24C0:EA A9 0A 85 C9 4C 6F 22<AD>
24C8:A9 15 4C 3D E9 A9 00 AA<1B>
24D0:9D 00 D8 9D 00 D9 9D 00<58>
24D8:DA 9D 00 DB 9D 00 DC 9D<63>
24E0:00 DD E8 D0 EB A5 84 85<D3>
24E8:8E 85 90 85 0E A5 85 85<63>
24F0:8F 85 91 85 0F 4C 86 F0<50>
24F8:85 89 18 A5 90 85 88 65<3E>
2500:89 A8 A5 91 85 89 69 00<6F>
2508:CD E6 02 90 09 D0 10 CC<EF>
2510:E5 02 90 02 D0 09 85 91<BE>
2518:85 0F 84 90 84 0E 60 A9<B8>
2520:02 4C 3D E9 A5 8F C5 91<12>
2528:90 06 A5 8E C5 90 B0 25<2A>
2530:A5 90 E9 02 85 90 85 0E<A9>
2538:B0 04 C6 91 C6 0F A0 00<55>
2540:B1 90 D0 10 38 A5 90 E9<A4>
2548:0D 85 90 85 0E B0 04 C6<B5>
2550:91 C6 0F 98 18 60 20 66<18>
2558:FA B0 04 F0 F9 10 11 A9<DE>
2560:1C 2C A9 10 4C 3D E9 20<02>
2568:66 FA B0 F6 F0 F9 10 F2<4F>
2570:A0 02 68 68 B1 90 48 88<7E>
2578:B1 90 48 60 A2 01 2C A2<39>
2580:FF A9 03 20 3A FA A0 00<08>
2588:8A 91 88 C8 68 AA 18 69<F0>
2590:03 91 88 C8 68 48 69 00<4D>
2598:91 88 8A 48 60 85 BC 84<E7>
25A0:BD 60 20 4A F0 A9 03 20<52>
25A8:3A FA A0 00 A9 FF 91 88<F4>
25B0:C8 68 18 69 03 91 88 C8<B5>
25B8:68 69 00 91 88 6C D8 00<0F>
25C0:A6 14 A5 13 A4 12 E4 14<91>
25C8:D0 F6 86 D3 20 13 C0 A9<4B>
25D0:89 A0 FB 20 0B F2 20 B0<38>
25D8:C2 A5 D3 A0 00 20 13 C0<27>
25E0:4C D1 C0 20 02 FB A9 8F<7C>
25E8:A0 FB 20 14 F2 A0 00 A5<B9>
25F0:D4 C9 40 D0 02 84 D5 84<A7>
25F8:F2 A9 24 20 51 FB A9 60<15>
2600:20 51 FB A9 60 20 51 FB<CB>
2608:A0 06 A9 80 4C 51 F4 48<AA>
2610:20 EC C1 20 B0 C2 20 DD<3B>
2618:C3 68 85 D5 20 0B E4 A9<6B>
2620:00 A4 D4 C0 40 D0 05 A4<BB>
2628:D5 85 D5 98 AA 4A 4A 4A<89>
2630:4A 20 79 FB 8A 29 0F 09<28>
2638:30 C9 3A 90 02 69 06 A4<80>
2640:F2 99 80 05 E6 F2 60 41<43>
2648:02 56 00 00 00 43 04 32<6E>
2650:00 00 00 20 84 F3 84 E1<70>
2658:84 E2 20 DE FB C9 18 B0<FC>
2660:35 85 E0 20 2D FC 20 DE<18>
2668:FB C9 3C B0 29 20 F8 FB<93>
2670:20 2D FC 20 DE FB C9 3C<42>
2678:B0 1C 20 F8 FB 20 0C FC<93>
2680:20 06 FC A5 E0 A4 E1 A6<25>
2688:E2 85 14 84 13 86 12 C5<79>
2690:14 D0 F6 4C D0 F3 20 D0<B8>
2698:F3 4C 9F F3 20 09 C0 E6<FF>
26A0:F2 B0 F3 0A 85 E3 0A 0A<A6>
26A8:65 E3 85 E3 20 09 C0 E6<39>
26B0:F2 B0 E3 65 E3 60 18 65<CE>
26B8:E0 85 E0 90 06 E6 E1 D0<F4>
26C0:02 E6 E2 60 06 E0 26 E1<A5>
26C8:26 E2 A4 E2 A5 E0 A6 E1<0B>
26D0:06 E0 26 E1 26 E2 06 E0<8E>
26D8:26 E1 26 E2 65 E0 85 E0<E6>
26E0:8A 65 E1 85 E1 98 65 E2<11>
26E8:85 E2 60 20 06 FC A4 E2<C6>
26F0:A5 E0 A6 E1 06 E0 26 E1<46>
26F8:26 E2 65 E0 85 E0 8A 65<A0>
2700:E1 85 E1 98 65 E2 06 E0<0C>
2708:26 E1 2A 85 E2 60 20 75<83>
2710:FC 86 D3 38 E9 01 C0 00<A6>
2718:D0 29 C5 D3 B0 25 0A A8<38>
2720:68 85 D4 68 85 D5 C8 B1<10>
2728:D4 85 D6 C8 B1 D4 85 D7<95>
2730:6C D6 00 86 D3 20 F9 CA<D5>
2738:B0 07 A5 D4 A4 D5 A6 D3<29>
2740:60 68 68 06 D3 68 65 D3<BB>
2748:A8 68 69 00 48 98 48 60<87>
2750:20 75 FC 18 90 04 20 75<0E>
2758:FC 38 08 86 D3 38 E9 01<66>
2760:C0 00 D0 DE C5 D3 B0 DA<15>
2768:AA A9 03 20 3A FA A0 00<53>
2770:A9 FF 28 6A 91 88 C8 06<B6>
2778:D3 68 85 D4 65 D3 91 88<11>
2780:C8 68 85 D5 69 00 91 88<7E>
2788:8A 0A A8 4C 68 FC A5 D7<23>
2790:A4 D6 85 DB 84 DA A0 00<49>
2798:84 DC 84 DD A0 10 46 D9<90>
27A0:66 D8 90 0F 18 A5 DC 65<EA>
27A8:DA 85 DC A5 DD 65 DB 85<92>
27B0:DD 30 0E 06 DA 26 DB 88<40>
27B8:D0 E4 A5 DD C9 16 B0 01<EA>
27C0:60 4C 81 F0 86 D3 20 3E<21>
27C8:EB A9 00 A8 A6 D3 86 D3<D0>
27D0:18 69 01 85 D8 48 98 69<32>
27D8:00 30 E6 85 D9 48 C6 92<3C>
27E0:A4 92 18 B9 00 DE 69 01<7E>
27E8:85 D6 B9 2A DE 69 00 30<F0>
27F0:D0 85 D7 20 D0 FC 06 DC<7A>
27F8:26 DD A4 DD A5 DC 0A 26<A9>
2800:DD 65 DC 85 DC 98 65 DD<90>
2808:85 DD 68 85 D9 68 85 D8<70>
2810:4C 6C FD 86 D3 85 D8 84<0D>
2818:D9 85 DC 84 DD 05 D9 F0<96>
2820:A0 98 30 9D A9 00 85 D6<5A>
2828:85 D7 A6 D3 BD 00 D9 D0<BB>
2830:90 A5 8E 85 D4 A5 8F 85<0F>
2838:D5 A5 DC A4 DD 20 DF EC<63>
2840:A6 D3 20 51 EB A9 00 A8<47>
2848:A6 DD F0 0A 91 D4 C8 D0<FD>
2850:FB E6 D5 CA D0 F6 A6 DC<5C>
2858:F0 06 91 D4 C8 CA D0 FA<98>
2860:60 84 E7 A0 00 84 E8 84<1E>
2868:E9 F0 10 85 E8 84 E9 C6<58>
2870:92 A4 92 B9 2A DE 85 E7<29>
2878:B9 00 DE 85 E6 20 77 EB<C2>
2880:A5 E8 C5 D8 A5 E9 E5 D9<A2>
2888:B0 35 A4 E6 C4 D6 A5 E7<80>
2890:E5 D7 B0 2B A5 E7 20 D4<9E>
2898:FC 18 A5 DC 65 E8 85 DC<F9>
28A0:A5 DD 65 E9 06 DC 2A 85<4E>
28A8:DD A8 A5 DC 0A 26 DD 65<A7>
28B0:DC AA 98 65 DD 65 D5 A8<E0>
28B8:8A 65 D4 90 01 C8 60 4C<5C>
28C0:81 F0 20 A3 FD 4C B5 EB<76>
28C8:20 AD FD 4C B5 EB 86 D3<E7>
28D0:20 13 C0 A6 D3 86 D3 20<BD>
28D8:F5 EA A5 11 F0 34 A5 D4<30>
28E0:A4 D5 A6 D3 20 A3 FD 4C<2A>
28E8:5F EC 86 D3 20 13 C0 A6<55>
28F0:D3 86 D3 20 F5 EA A5 11<11>
28F8:F0 18 A5 D4 A4 D5 A6 D3<D0>
2900:20 AD FD 4C 5F EC 20 04<D0>
2908:FE 4C CE ED 20 0A FE 4C<66>
2910:CE ED A9 80 4C 3D E9 86<60>
2918:D3 A6 11 F0 F5 A6 D4 86<0A>
2920:88 A6 D5 86 89 A6 D6 86<A7>
2928:8C A6 D7 86 8D A6 D3 4C<1B>
2930:77 EB 20 59 FE A5 D9 A4<5A>
2938:D8 20 AF FE 9D 00 DB 98<E7>
2940:9D 00 DA 60 20 59 FE 20<31>
2948:01 FF 20 AB FE 4C 7E FE<A5>
2950:20 59 FE 20 E0 FE 20 AB<6C>
2958:FE DD 00 DB 90 0A D0 DC<BE>
2960:48 98 DD 00 DA 68 B0 D4<40>
2968:60 A5 D7 A4 D6 C5 8D 90<9C>
2970:06 D0 08 C4 8C B0 04 85<DF>
2978:8D 84 8C 18 A5 D4 85 8A<2C>
2980:65 8C 85 E0 A5 D5 85 8B<E3>
2988:65 8D 85 E1 20 26 22 38<2C>
2990:A6 D3 A5 E0 FD 00 D8 A8<5C>
2998:A5 E1 FD 00 D9 60 C6 92<76>
29A0:A6 92 BD 00 DE 85 E2 BD<87>
29A8:2A DE 85 E3 05 E2 F0 0A<7F>
29B0:A5 D8 C5 E2 A5 D9 E5 E3<C0>
29B8:B0 0D A9 05 4C 3D E9 A5<F9>
29C0:D8 85 E2 A5 D9 85 E3 C6<8E>
29C8:92 A6 92 38 BD 00 DE E9<1D>
29D0:01 85 E0 BD 2A DE E9 00<46>
29D8:85 E1 90 DE A5 E2 E5 E0<6B>
29E0:85 D6 A5 E3 E5 E1 85 D7<DC>
29E8:90 D0 05 D6 F0 CC 18 A5<B8>
29F0:D4 65 E0 85 D4 A5 D5 65<DD>
29F8:E1 85 D5 60 85 E2 84 E3<14>
2A00:05 E3 F0 B6 20 77 EB A5<42>
2A08:D6 C5 E2 A5 D7 E5 E3 B0<0F>
2A10:B6 90 A7 48 20 77 EB A6<F7>
2A18:D6 86 E2 A6 D7 86 E3 68<9A>
2A20:38 E9 01 85 E0 98 4C 18<C0>
2A28:FF 85 D4 A9 21 48 48 86<5E>
2A30:D6 86 D7 8A F0 16 A6 92<7B>
2A38:A5 D4 48 98 48 CA BD 00<B3>
2A40:DE BC 2A DE C6 D6 D0 F2<F0>
2A48:85 D4 86 92 84 D5 A5 D7<BD>
2A50:48 4C 17 21 A5 C3 A0 00<38>
2A58:4C 13 C0 A5 BA A4 BB 4C<01>
2A60:13 C0 86 D4 18 65 D4 90<CA>
2A68:03 C8 F0 0D 60 86 D4 38<F1>
2A70:E5 D4 B0 F8 88 C0 FF D0<B2>
2A78:F3 4C 29 F5 00 00 00 00<CD>
Laenge 10876 Bytes
```

Listing 2. Turbo-Basic-Compiler-Runtime-Paket (Schluß)

Von der Ordnung zum Chaos – Apfelmännchen auf dem Atari

Faszinierende Grafiken von nahezu unendlicher Vielfalt erzeugt das Programm »Apfelmännchen« auf dem Atari 800XL in Turbo-Basic.

Grundlage dieses Programms ist ein Teilgebiet der Mathematik von großer Komplexität. Es handelt sich um die Berechnung und Betrachtung der sogenannten Mandelbrot-Menge und speziell deren Grenzschichten. Diese Menge, benannt nach dem Wissenschaftler Benoit B. Mandelbrot, besteht aus einer Folge von komplexen Zahlen, die sich recht merkwürdig verhalten. Trägt man die beiden Komponenten jeder komplexen Zahl, die sich aus diesem Algorithmus ergibt, auf die Abszisse und die Ordinate eines Koordinatensystems auf, so erhält man Grafiken, die kuriose Eigenschaften aufweisen.

Um den grundlegenden Algorithmus zu verstehen, muß man sich etwas näher mit den Eigenschaften komplexer Zahlen auseinandersetzen. Eine komplexe Zahl besteht immer aus zwei Komponenten, dem Realteil und dem Imaginärteil. Die Schreibweise erfolgt in der Form: »C=a+bi«. Die komplexe Zahl C ergibt sich also aus dem Realteil a und dem Imaginärteil b. Der Buchstabe i steht für eine Konstante und unterliegt folgender Definition: »$i^2=-1$«. Die Rechenregeln der komplexen Zahlen weichen jedoch von denen der reellen Zahlen ab. Geht man bei der Addition noch in gewohnter Weise vor und addiert Real- und Imaginärteil einfach separat, ist das Multiplizieren schon etwas komplizierter. So ergibt sich »(a+bi)*(c+di)« zu »(a*c-b*d)+(a*d+b*c)*i«. Wer diese Formel genauer betrachtet, wird feststellen, daß sich »i^2« aufgrund seiner Definition zu -1 aufgelöst hat.

Beim »Apfelmännchen« geht man von der komplexen Zahl Null aus, zieht eine komplexe Zahl C ab und quadriert das Ergebnis. Nun zieht man erneut C ab und quadriert das Ganze wiederum. Dieser Vorgang wiederholt sich so lange, bis ein vorher festgelegtes Grenzkriterium überschritten wird. Alle Zahlen, die hierbei gegen Unendlich streben, gehören zur Mandelbrot-Menge. Am Rand dieser Menge stellt sich jedoch ein seltsamer Effekt ein. Hier fallen oder steigen die Zahlen bei wechselndem Vorzeichen scheinbar völlig regellos. Dies ist der für die Grafik interessante Teil der Mandelbrot-Menge. Zwei Grenzkriterien legen nun fest, ob man sich innerhalb oder außerhalb beziehungsweise am Rand der Mandelbrot-Ebene befindet. Das erste Kriterium bricht den Algorithmus nach einer vorher festgelegten Zahl von Durchgängen ab. Im Listing geschieht dies durch die Variable »TMAX«. Wird dieser Wert erreicht, nimmt man an, daß die Folge gegen Unendlich strebt. Der Punkt auf dem Bildschirm erhält die Farbe Schwarz. Im anderen Fall wurde das zweite Grenzkriterium zuerst erreicht. Dies ist dann der Fall, wenn die komplexe Zahl einen bestimmten Wert überschreitet. Im Listing wurde der Grenzwert 8 angenommen, der jedoch keine feste Größe darstellt, und fast beliebig geändert werden kann. Die zuzuordnende Farbe wird nun von der Anzahl der Iterationen abhängig gemacht. Im Listing geschieht dies mit der Zuordnung »TIEFE MOD 3+1«. Diese Berechnung kann zu den Farben Eins, Zwei und Drei, niemals jedoch zu der Farbe Null führen.

Für jeden Punkt, der berechnet werden soll, wird natürlich eine andere komplexe Zahl C gewählt. Die Umgebung der Mandelbrot-Ebene ergibt sich für folgende Werte: Realteil von -2 bis 1 und Imaginärteil von -1.5 bis 1.5. Innerhalb dieser Grenzen kann man nun beliebig kleine Ausschnitte auf dem Bildschirm darstellen, also Teilbereiche der Menge gewissermaßen vergrößern. Dabei stellt man fest, daß immer neue Grafiken oder Figuren entstehen, die zum Teil zu anderen Ausschnitten ähnlich, aber nie genau gleich sind. Die einzige Einschränkung, die hierbei vom Computer auferlegt wird, liegt in der begrenzten Rechengenauigkeit. Bei theoretisch unendlicher Rechengenauigkeit ließe sich die Grafik unendlich oft vergrößern. Geht man also beispielsweise von den Grenzen -1,1,-1 und 1 aus, so wären als nächste Vergrößerungsstufe die Werte -0.5,0.5,-0.5 und 0,5 denkbar. Halbiert man diese Werte immer wieder, so ist klar ersichtlich, daß auch eine zehnstellige Rechengenauigkeit bald nicht mehr zur Darstellung der Zahlen genügt.

Komplexe Formel, schöne Grafik

Zur Realisierung dieses Programms auf dem Atari wurde als Programmiersprache Turbo-Basic XL gewählt. Die zirka viermal so hohe Rechengeschwindigkeit, die diese Sprache gegenüber dem normalen Basic aufweist, ist für eine akzeptable Rechendauer unbedingt notwendig. Die Grafik wird in Grafikstufe 15 erzeugt. Dies bedeutet, daß genau 30720 Bildpunkte berechnet werden müssen. Abhängig von der maximalen Iterationszahl, die man mit »TMAX« bestimmt hat, kann jeder Punkt zudem bis zu 250mal berechnet werden. Selbst unter Turbo-Basic XL führt dies in Extremfällen zu Rechenzeiten von acht Stunden und mehr. Die Programmierung in Atari-Basic erscheint also wenig sinnvoll, will man nicht ein bis zwei Tage auf die Vollendung einer Grafik warten.

Handhabung des Programms

Nach dem Starten des Programms kommt man in das zentrale Menü, von dem aus alle Funktionen direkt ansprechbar sind. Punkt eins des Menüs betrifft den Start eines Apfelmännchens. Man muß zuerst die obere und untere Grenze des Real- und Imaginärbereichs eingeben. Auf die Frage

PROGRAMM-STECKBRIEF	
Programmname	Apfelmännchen
Programmtyp	Grafik
Programmiersprache	Turbo-Basic XL
Programmlänge	7155 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	Prüfsummer
Bemerkung	Erzeugt und speichert Grafiken, die mittels einer speziellen mathematischen Formel berechnet werden. Auf der Leserservice-Diskette sind einige fertige Grafiken bereits gespeichert.
Leserservice	Diskette (APFELTUR)

»TIEFE?« antwortet man mit der maximalen Anzahl der Iterationen, die man zur Berechnung jedes Punktes zulassen will. Je höher man diesen Wert wählt, desto feiner werden die Grenzbereiche zur Mandelbrot-Ebene berechnet. Dies bedeutet allerdings auch eine Erhöhung der Rechenzeit, so daß man hier meist Kompromisse eingehen wird. Die Berechnung der Grafik kann dann auf dem Bildschirm mitverfolgt werden. Nach der Fertigstellung springt das Programm zurück ins Hauptmenü. Das fertige Bild kann man nun mit Option 2 auf dem Bildschirm betrachten. Mit den Tasten 0, 1, 2, und 3 werden die vier Grundfarben nach Belieben nachträglich geändert. Mit Hilfe des dritten Menüpunktes speichert man das fertige Bild. Wie es auch im Micropainterformat üblich ist, werden am Ende des Bildes vier Byte angehängt, die Auskunft über die verwendeten Farben geben. Zusätzlich speichert das Programm allerdings noch die eingegebenen Parameter. So kann man mit Punkt vier des Menüs also auch nur solche Bilder von Diskette laden, die das eben beschriebene Format besitzen.

Apfelmännchen schön und bunt

Das Inhaltsverzeichnis einer Diskette erhält man mit Punkt fünf des Menüs. Der sechste Punkt, nämlich »Effekte«, läßt die drei Vordergundfarben eines Apfelmännchens über den Bildschirm »fließen«. Dies gibt der Grafik je nach Aussehen interessante, teilweise räumlich wirkende Effekte.

Um das Vergrößern einzelner Bildteile zu erleichtern, wurde Menüpunkt sieben aufgenommen. Mit Hilfe eines Joysticks kann man einen Bildausschnitt eingrenzen und erneut berechnen lassen. Dies geschieht, indem man zwei Eckpunkte eines gedachten Rechtecks entsprechend auf dem Bildschirm plaziert. Welchen der Eckpunkte man bewegen will, wird mit der Feuertaste gewählt. Die Berechnung der neuen Grafik beginnt nach dem Drücken der START-Taste. Will man sich die neuen Parameter vorher noch ansehen, so wählt man einfach OPTION. Funktion acht des Menüs zeigt nun die Parameter, und mit Punkt neun startet die Berechnung des neuen Apfelmännchens.

Zum Programmaufbau

Das Programm »Apfelmännchen« wurde in Turbo-Basic XL geschrieben und konnte daher nicht nur wesentlich schneller, sondern auch strukturierter als im normalen Basic angelegt werden. So wird es auf den ersten Blick auffallen, daß beispielsweise auf Zeilennummern-orientierte Sprunganweisungen mit Ausnahme der »TRAP«-Befehle vollständig verzichtet wurde. Alle Hauptprogrammteile und Unterroutinen sind als Prozeduren angelegt und sinngemäß benannt.

Die Hauptschleife besteht aus einer DO-LOOP-Anweisung von Zeile 95 bis Zeile 240. Hier wird das Hauptmenü auf den Bildschirm gebracht, und von hier aus erfolgt der Sprung zu allen wesentlichen Prozeduren. Die einzelnen Prozeduren:

DECLARE

Es erfolgt die Dimensionierung aller Stringvariablen sowie die Erstbelegung der Variablen für die Änderung der Farbregister. Das Aussehen der beiden Player, die für die Ausschnittberechnung benötigt werden, legen die Variablen LO$ und RU$ fest.

RECHNEN

Hier findet man den Berechnungsalgorithmus, wie er im wesentlichen bereits erklärt wurde. Verwendung findet die Grafikstufe 15; man verfügt also über vier Farben bei einer Auflösung von 160 mal 192 Bildpunkten. Es wäre auch eine andere Grafikstufe denkbar. In diesem Fall müßten folgende Änderungen vorgenommen werden: Anpassung der Variablen DX, DY sowie der Schleifen ZEILE und SPALTE an die neue Auflösung. In Zeile 1230 muß anstatt der drei die Anzahl der Farben minus eins eingesetzt werden.

EFFEKTE

Die drei Farbregister für die Vordergrundfarben werden mit verschiedenen Farbwerten belegt. Durch die Verschiebung der Farben entsteht der Eindruck des Fließens.

PARAMETER

Hier gibt man die Randwerte für ein neues Apfelmännchen ein. Ist dies geschehen, wird die Prozedur RECHNEN aufgerufen.

SEHEN

Über die Prozedur BILDHOLEN wird das momentan im Speicher befindliche Apfelmännchen auf dem Bildschirm dargestellt. Die Farbregister 708, 709, 710 und 712 werden durch Drücken der Tasten 0 bis 3 jeweils um den Wert eins inkrementiert. Die REPEAT-Anweisung wird bis zum Betätigen der Leertaste ausgeführt.

SPEICHERN

Zuerst erfolgt der Aufruf der Prozedur »FILEGET«. Dabei wird der Filename des Bildes gelesen, auf Fehler überprüft und gegebenenfalls ergänzt. Mit dem BPUT-Befehl wird der Inhalt des Bildschirmspeichers auf Diskette abgelegt. Es folgen die vier Farbbytes und die Parameter. Da es sich bei letzteren um Gleitkommazahlen handelt, werden sie als Strings gespeichert.

LADEN

Diese Prozedur entspricht fast vollständig der Prozedur »SPEICHERN«. Alle Schreibbefehle sind sinngemäß in Lesebefehle umgewandelt. Eventuell auftretende Fehler bei der Diskettenhandhabung werden von TRAP-Routinen abgefangen.

DIRECTORY

Der linke Bildrand wird auf Position 7 gesetzt und das Directory mit dem DIR-Befehl aufgerufen. Nach Betätigung einer Taste erfolgt der Rücksprung ins Hauptmenü.

Ausschnitt

Diese Prozedur setzt sich aus mehreren Unterprogrammen zusammen. PMINIT initialisiert die grundsätzlichen Voreinstellungen für den Player/Missile-Aufruf. Verwendung finden die Player null und eins. Sie werden in Einzelzeilen-Auflösung dargestellt. Ihre Farbe ist Weiß und die erste Position wird durch die Variablen XO, YO, X1 und Y1 definiert. PMBUILD positioniert die beiden Player auf dem Bildschirm. Der Aufbau erfolgt dabei durch den MOVE-Befehl. Die folgende REPEAT-Anweisung fragt den Joystick und die Sondertasten ab. Abhängig von der Auslenkrichtung des Joysticks werden die Prozeduren »PMLEFT«, »PMRIGHT«, »PMUP« und »PMDOWN« angesprungen. Sie versetzen den jeweiligen Player um einen Pixel in die gewünschte Richtung. Ist die Positionierung abgeschlossen, werden die alten Parameter durch die neuen Werte ersetzt. Die Prozedur »PMEXIT« schaltet zuletzt die beiden Player wieder aus.

PARAMANZ

Hier werden lediglich die aktuellen Parameter auf dem Bildschirm angezeigt. Nach dem Betätigen einer beliebigen Taste gelangt man wieder ins Hauptmenü.

(Wolfgang Czerny/wb)



```
10 --                                         <YW>
15 REM - Apfelmaennchen in          -         <DT>
17 REM - Turbo-Basic XL fuer        -         <FU>
19 REM - Atari-Computer mit         -         <CE>
20 REM - mindestens 48 KByte-RAM-             <BK>
25 REM -                            -         <KM>
27 REM - Programmiert von           -         <RA>
29 REM - Wolfgang Czerny            -         <AR>
30 REM - (c) Happy-Computer         -         <LL>
35 --                                         <ZI>
90 EXEC DECLARE                               <YX>
95 DO                                         <RA>
100 GRAPHICS 0:POKE 752,1                     <GU>
110 POKE 709,0:POKE 710,10:POKE 712,10        <BY>
115 POKE 82,0                                 <HV>
120 POSITION 9,1:? "HAUPTMENUE"               <XC>
130 POSITION 9,2:? "----------"               <BC>
140 POSITION 3,5:? "(1)___Parameter_eing
eben"                                         <SM>
150 POSITION 3,7:? "(2)___Bild_ansehen"       <AW>
160 POSITION 3,9:? "(3)___Bild_speichern
"                                             <DV>
170 POSITION 3,11:? "(4)___Bild_laden"        <WQ>
180 POSITION 3,13:? "(5)___Directory"         <IU>
190 POSITION 3,15:? "(6)___Effekte"           <GC>
195 POSITION 3,17:? "(7)___Auschnitt"         <JF>
197 POSITION 3,19:? "(8)___Parameter_anz
eigen"                                        <WK>
199 POSITION 3,21:? "(9)___Rechnen"           <RF>
200 POSITION 3,23:? "(0)___Ende";             <EA>
205 REPEAT                                    <DM>
210 GET EIN                                   <HE>
220 UNTIL EIN>47 AND EIN<58                   <OH>
225 IF EIN=48 THEN EXIT                       <EF>
230 ON EIN-48 EXEC PARAMETER,SEHEN,SPEIC
HERN,LADEN,DIRECTORY,EFFEKTE,AUSSCHNITT,
PARAMANZ,RECHNEN                              <XD>
240 LOOP                                      <CF>
250 END                                       <XX>
1000 --                                       <VE>
1002 REM -    Rechenroutine         -         <IU>
1003 --                                       <VQ>
1009 PROC RECHNEN                             <NR>
1010 GRAPHICS 15+16:POKE 764,255              <QS>
1030 DX=(XMAX-XMIN)/159                       <MN>
1040 DY=(YMAX-YMIN)/191                       <KE>
1050 CX=XMIN:CY=YMAX                          <SU>
1080 FOR ZEILE=0 TO 191                       <FH>
1090 FOR SPALTE=0 TO 159                      <WH>
1110 TI=0:XW=0:YW=0:X2=0:Y2=0                 <JY>
1120 WHILE TI<TMAX AND (X2+Y2)<8              <PY>
1130 YW=2*XW*YW-CY:XW=X2-Y2-CX                <TU>
1150 X2=XW^2:Y2=YW^2:TI=TI+1                  <XN>
1180 WEND                                     <UE>
1190 IF TI=TMAX                               <FI>
1200 COLOR %0                                 <QL>
1220 ELSE                                     <TE>
1230 COLOR (TI MOD 3)+%1                      <ID>
1240 ENDIF                                    <SW>
1250 CX=CX+DX:PLOT SPALTE,ZEILE               <XA>
1260 IF PEEK(764)=28 THEN POP :GOTO 1295      <FH>
1270 NEXT SPALTE                              <DB>
1280 CX=XMIN:CY=CY-DY                         <DC>
1290 NEXT ZEILE                               <GB>
1295 EXEC BILDSPEICHERN                       <WL>
1300 ENDPROC                                  <OI>
1301 --                                       <VO>
1302 REM -     Bildeffekte          -         <DV>
1303 --                                       <VW>
1305 PROC EFFEKTE                             <JT>
1310 EXEC BILDHOLEN                           <IF>
1370 POKE 764,255                             <GS>
1375 REPEAT                                   <FU>
1380 POKE 708,COL0:POKE 709,COL1              <QE>
1390 POKE 710,COL2:PAUSE 5                    <HI>
1400 POKE 708,COL1:POKE 709,COL2              <RG>
1410 POKE 710,COL0:PAUSE 5                    <FC>
1420 POKE 708,COL2:POKE 709,COL0              <PS>
1430 POKE 710,COL1:PAUSE 5                    <GA>
1455 UNTIL PEEK(764)<>255                     <XF>
1460 ENDPROC                                  <PC>
1470 --                                       <WH>
1480 REM -   Parametereingabe       -         <MK>
1490 --                                       <WN>
2000 PROC PARAMETER                           <UK>
2005 REPEAT                                   <EU>
2010 TRAP 2010                                <LY>
2020 CLS :POKE 752,0                          <NK>
2030 POSITION 9,3:? "PARAMETER_EINGEBEN"      <WB>
2040 POSITION 9,4:? "------------------"      <YI>
2045 REPEAT                                   <FG>
2050 POSITION 3,7:INPUT "Linker_Rand_=_"
;XMIN                                         <SZ>
2060 POSITION 3,9:INPUT "Rechter_Rand_=_
";XMAX                                        <HT>
2065 UNTIL XMIN<XMAX                          <PB>
2067 REPEAT                                   <FU>
2070 POSITION 3,11:INPUT "Unterer_Rand_=
_";YMAX                                       <SS>
2080 POSITION 3,13:INPUT "Oberer_Rand_=_
";YMIN                                        <WL>
2082 UNTIL YMAX<YMIN                          <PO>
2083 REPEAT                                   <FK>
2085 POSITION 3,15:INPUT "Tiefe_(5_-_254
)_=";TMAX                                     <DJ>
2087 UNTIL TMAX>4 AND TMAX<255                <WJ>
2090 POSITION 3,20:? "ALLES_IN_ORDNUNG_?
_(J/N)";                                      <DO>
2091 REPEAT                                   <FF>
2092 GET EIN                                  <JJ>
2093 UNTIL EIN=74 OR EIN=78                   <ZR>
2094 UNTIL EIN=74                             <KG>
2095 EXEC RECHNEN                             <KD>
2100 ENDPROC                                  <OF>
2110 --                                       <VK>
2120 REM -      Bild ansehen        -         <DP>
2130 --                                       <VQ>
3000 PROC SEHEN                               <YF>
3010 EXEC BILDHOLEN                           <IB>
3015 REPEAT                                   <EY>
3017 REPEAT                                   <FG>
3020 GET EIN                                  <IH>
3040 UNTIL (EIN>47 AND EIN<52) OR EIN=32      <UQ>
3050 IF EIN-48=0 THEN COL0=COL0+1:IF COL
0>255 THEN COL0=COL0-256                      <ZQ>
3060 IF EIN-48=1 THEN COL1=COL1+1:IF COL
1>255 THEN COL1=COL1-256                      <IO>
3070 IF EIN-48=2 THEN COL2=COL2+1:IF COL
2>255 THEN COL2=COL2-256                      <RM>
3080 IF EIN-48=3 THEN COL3=COL3+1:IF COL
3>255 THEN COL3=COL3-256                      <AK>
3090 POKE 708,COL0:POKE 709,COL1:POKE 71
0,COL2:POKE 712,COL3                          <XJ>
3100 UNTIL EIN=32                             <EY>
3110 EXEC BILDSPEICHERN                       <UT>
3900 ENDPROC                                  <OW>
3990 --                                       <WZ>
3992 REM -   Bild auf Diskette      -         <KC>
3993 REM -       speichern          -         <GS>
3994 --                                       <XP>
4000 PROC SPEICHERN                           <SV>
4010 CLS                                      <TL>
4020 POSITION 9,3:? "BILD_SPEICHERN"          <BX>
4030 POSITION 9,4:? "--------------"          <TL>
4050 EXEC FILEGET                             <IN>
4100 EXEC BILDHOLEN                           <IB>
4110 OPEN #1,8,0,DAT$                         <SV>
4120 BPUT #1,DPEEK(88),7680                   <EU>
4130 PUT #1,COL0,COL1,COL2,COL3               <PH>
4140 PRINT #1,STR$(XMIN)                      <DJ>
4150 PRINT #1,STR$(XMAX)                      <FO>
4160 PRINT #1,STR$(YMIN)                      <EJ>
```

Turbo-Basic XL-Listing zu »Apfelmännchen«

```
4170 PRINT #1,STR$(YMAX)                         <GO>
4180 PRINT #1,STR$(TMAX)                         <CV>
4190 CLOSE #1                                    <NP>
4900 ENDPROC                                     <OX>
4990 --                                          <XA>
4992 REM -      Bild von Diskette      -         <QJ>
4993 REM -             laden           -         <ZH>
4994 --                                          <XQ>
5000 PROC LADEN                                  <RM>
5010 CLS                                         <TM>
5020 POSITION 9,3:? "BILD_LADEN"                 <IG>
5030 POSITION 9,4:? "----------"                 <QI>
5050 EXEC FILEGET                                <IO>
5100 GRAPHICS 15+16                              <FR>
5110 OPEN #1,4,0,DAT$                            <QS>
5120 BGET #1,DPEEK(88),7680                      <XM>
5125 GET #1,COL0,COL1,COL2,COL3                  <JP>
5130 INPUT #1,E$:XMIN=VAL(E$)                    <GM>
5140 INPUT #1,E$:XMAX=VAL(E$)                    <IN>
5150 INPUT #1,E$:YMIN=VAL(E$)                    <HK>
5160 INPUT #1,E$:YMAX=VAL(E$)                    <JL>
5165 INPUT #1,E$:TMAX=VAL(E$)                    <GT>
5170 CLOSE #1                                    <NK>
5180 EXEC BILDSPEICHERN                          <VQ>
5190 ENDPROC                                     <PJ>
5199 --                                          <XV>
5200 REM -    Directory auf dem      -           <PR>
5210 REM -    Bildschirm ausgeben    -           <PI>
5220 --                                          <VS>
6000 PROC DIRECTORY                              <JQ>
6010 CLS :POKE 82,7:?                            <XQ>
6020 DIR                                         <ST>
6030 GET EIN                                     <IN>
6035 POKE 82,0                                   <LP>
6040 ENDPROC                                     <OT>
6050 --                                          <VY>
6060 REM -    Filename abfragen      -           <VX>
6070 --                                          <WE>
7000 PROC FILEGET                                <MB>
7005 TRAP 7005                                   <RC>
7007 POKE 752,0                                  <PC>
7010 POSITION 3,7:INPUT "WELCHER_FILE-NA
ME_";FILE$                                       <JL>
7020 DAT$="_______________"                      <AO>
7050 IF FILE$(1,2)<>"D:"                         <EN>
7060 DAT$(3,LEN(FILE$)+2)=FILE$                  <TB>
7070 DAT$(1,2)="D:"                              <CH>
7080 ELSE                                        <TY>
7090 DAT$=FILE$                                  <UD>
7095 ENDIF                                       <UH>
7110 TRAP 20000                                  <NK>
7900 ENDPROC                                     <PA>
7990 --                                          <XD>
7992 REM -   Ausschnitt waehlen      -           <MB>
7993 REM -   fuer diesen Programm-   -           <PA>
7994 REM -   teil wird ein Joy-      -           <YA>
7995 REM -   stick benoetigt         -           <VL>
7996 --                                          <YB>
8000 PROC AUSSCHNITT                             <QS>
8010 EXEC BILDHOLEN                              <IG>
8015 EXEC PMINIT                                 <ZW>
8020 EXEC PMBUILD                                <NA>
8022 REPEAT                                      <EU>
8024 REPEAT                                      <FC>
8030 JOY=STICK(0)                                <CP>
8040 PRESS=STRIG(0)                              <FG>
8045 START=PEEK(53279)                           <IL>
8050 UNTIL JOY<>15 OR PRESS=0 OR START<>
7                                                <NQ>
8060 IF JOY=11 THEN EXEC PMLEFT                  <AA>
8070 IF JOY=7 THEN EXEC PMRIGHT                  <EO>
8080 IF JOY=14 THEN EXEC PMUP                    <YX>
8090 IF JOY=13 THEN EXEC PMDOWN                  <QL>
8100 IF PRESS=0 AND PMNR=0                       <MY>
8105 PMNR=1:SOUND 0,10,10,10:PAUSE 10            <PY>
8107 ELSE                                        <UF>
8110 IF PRESS=0 AND PMNR=1 THEN PMNR=0:S
OUND 0,100,10,10:PAUSE 10                        <UU>
8120 ENDIF                                       <SV>
8130 YDELTA=Y1POS-Y0POS                          <DQ>
8140 XDELTA=X1POS-X0POS                          <CH>
8145 DSOUND                                      <JO>
8150 UNTIL (START=6 OR START=5) AND XDEL
TA>1 AND YDELTA>1                                <TJ>
8160 DEX=203-40:DEY=223-32                       <DS>
8170 DXD=(XMAX-XMIN)/DEX                         <CP>
8180 DYD=(YMAX-YMIN)/DEY                         <EY>
8190 XMINN=XMIN+DXD*(X0POS-44)                   <VS>
8200 XMAXN=XMIN+DXD*(X1POS-35)                   <WR>
8210 YMINN=YMIN+DYD*(Y0POS-30)                   <RO>
8220 YMAXN=YMIN+DYD*(Y1POS-28)                   <BK>
8230 TMAX=254                                    <DC>
8240 XMIN=XMINN:XMAX=XMAXN                       <XZ>
8250 YMIN=YMINN:YMAX=YMAXN                       <AG>
8260 EXEC PMEXIT                                 <DF>
8270 IF START=6 THEN EXEC RECHNEN                <ZC>
8900 ENDPROC                                     <PB>
8990 --                                          <XE>
8991 REM -  Player Missile-Grafik -              <JV>
8992 REM -  initialisieren        -              <QJ>
8993 --                                          <XQ>
9000 PROC PMINIT                                 <DA>
9010 PMB=PEEK(106)-40:POKE 54279,PMB             <MV>
9020 PMB=PMB*256:POKE 559,62                     <XL>
9030 POKE 53256,0:POKE 53257,0                   <YU>
9040 PMNR=0:POKE 704,255:POKE 705,255            <HP>
9050 MOVE ADR(LEER$),PMB+1024,256                <BE>
9060 MOVE ADR(LEER$),PMB+1280,256                <FJ>
9070 X0POS=44:Y0POS=30                           <DF>
9080 X1POS=198:Y1POS=218                         <GF>
9090 ENDPROC                                     <PL>
9092 --                                          <WV>
9093 REM -  Player Missiles auf-  -              <FG>
9094 REM -  bauen und mani-       -              <KO>
9095 REM -  pulieren              -              <GK>
9099 --                                          <XX>
9100 PROC PMBUILD                                <QX>
9110 POKE 53277,3                                <EX>
9120 POKE 53248,X0POS:POKE 53249,X1POS           <RZ>
9130 MOVE ADR(LO$),PMB+1024+Y0POS,8              <NX>
9140 MOVE ADR(RU$),PMB+1280+Y1POS,8              <AC>
9190 ENDPROC                                     <FN>
9200 PROC PMEXIT                                 <GO>
9210 POKE 53277,0                                <DA>
9220 POKE 559,34                                 <YW>
9230 MOVE ADR(LEER$),PMB+1024,256                <BC>
9240 MOVE ADR(LEER$),PMB+1280,256                <FH>
9250 ENDPROC                                     <PD>
9290 --                                          <WR>
9292 REM -  Player Missiles nach  -              <NS>
9293 REM -  oben bewegen          -              <MD>
9294 --                                          <XH>
9300 PROC PMUP                                   <UN>
9310 IF PMNR=0                                   <VJ>
9312 Y0POS=Y0POS-1                               <UY>
9314 IF Y0POS<30 THEN Y0POS=30                   <XU>
9317 MOVE ADR(LO$),PMB+1024+Y0POS,8              <OX>
9318 ENDIF                                       <UD>
9320 IF PMNR=1                                   <WA>
9322 Y1POS=Y1POS-1                               <VV>
9324 IF Y1POS<28 THEN Y1POS=28                   <LR>
9327 MOVE ADR(RU$),PMB+1280+Y1POS,8              <BC>
9328 ENDIF                                       <UG>
9390 ENDPROC                                     <PR>
9392 --                                          <XB>
9393 REM -  Player Missiles nach  -              <NY>
9394 REM -  unten bewegen         -              <RN>
9399 --                                          <YD>
9400 PROC PMDOWN                                 <FT>
9410 IF PMNR=0                                   <VL>
9412 Y0POS=Y0POS+1                               <TS>
9414 IF Y0POS>221 THEN Y0POS=221                 <YA>
9416 MOVE ADR(LO$),PMB+1024+Y0POS,8              <OV>
9418 ENDIF                                       <UF>
9420 IF PMNR=1                                   <WC>
9422 Y1POS=Y1POS+1                               <UP>
9424 IF Y1POS>218 THEN Y1POS=218                 <KW>
9426 MOVE ADR(RU$),PMB+1280+Y1POS,8              <BA>
9428 ENDIF                                       <UI>
9590 ENDPROC                                     <PV>
```

Turbo-Basic XL-Listing zu »Apfelmännchen« **(Fortsetzung auf Seite 66)**

LOGO



"Buch des Jahres 1983" in den USA

Jeder kann programmieren
Computersprache für Eltern und Kinder
DANIEL WATT

LOGO ... Ergebnis der Erforschung menschlicher Intelligenz

Entwickelt von Seymour Papert, Pädagoge und Mathematikprofessor.
Erste Computersprache, die bewußt Strategien menschlichen Denkens dient – und in ihrer Logik der Realität gerecht wird.
LOGO ersetzt BASIC, sagen Pädagogen und Mathematiker.
LOGO kommt dem übergreifenden, assoziativen Denken entgegen. BASIC dagegen ist ein Setzkasten von Logik-Buchstaben.

DANIEL WATT ... hat im Team von Seymour Papert gearbeitet und ein Buch geschrieben, das voller Bilder seine Erlebnisse mit Kindern am Computer wiedergibt. Ein hochwertiges Textbuch für LOGO-Kurse. Ein Buch für Lehrer, die nach einem bereits von Schulbehörden empfohlenen LOGO-Kursbuch suchen.

Ein Buch für APPLE II, C-64, IBM PC, ATARI bis 520 ST., TI-99 und Schneider CPCs.
384 Seiten, A4, DM 59,–

te·wi

te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

COMPUTER FÜR KINDER



Ein Buch für Kinder und ihre Lehrer – ein kindgemäßes Buch für die erste Begegnung mit Computern, ihren Eigenwilligkeiten, und ihren unerschöpflichen Möglichkeiten. Ein Buch zu unserer Gegenwart und zur Zukunft unserer Kinder. „Computer für Kinder" richtet sich an Kinder im Alter von 8 bis 13 Jahren, für deren Interesse an Computern keines der unzähligen Computer-Bücher geschrieben wurde.

„Computer für Kinder" ist ganz auf Kinder eingestellt und beschäftigt sich unterhaltsam und leicht verständlich mit folgenden Themen:

Wie arbeiten Computer
Wie funktioniert mein Computer
Wie programmiert man mit einfachen Flußdiagrammen
Wie kann ich BASIC leicht verstehen
Programme aufbauen mit Befehlen
Farbige Graphiken entwerfen
Erklärung von Computer-Begriffen

Sally Greenwood Larson war Kindergärtnerin, ehe sie selbst Computern begegnete und zwischen den Welten von Kindern und Computern zu vermitteln begann.

Computer für Kinder, A4 quer, Fadenheftung, über 100 Seiten, je Ausgabe DM 29,80 vorliegend für: VC 20, C 64, Apple II, Atari

Fordern Sie Unterlagen über unser Gesamtprogramm an



DM 29.80



DM 29.80



DM 59,–

STRUCTURED BASIC FÜR COMMODORE 64

DM 199,–



DM 239,–



DM 49,–



DM 36,–

tm 4416/2

```
9592 --                                  <XF>
9593 REM -  Player Missiles nach  -      <OC>
9594 REM -  links bewegen         -      <NZ>
9599 --                                  <YH>
9600 PROC PMLEFT                         <YK>
9620 IF PMNR=0                           <VS>
9622 X0POS=X0POS-1                       <UP>
9624 IF X0POS<44 THEN X0POS=44           <FR>
9626 POKE 53248,X0POS                    <OZ>
9628 ENDIF                               <UM>
9630 IF PMNR=1                           <WJ>
9632 X1POS=X1POS-1                       <VM>
9634 IF X1POS<43 THEN X1POS=43           <FI>
9636 POKE 53249,X1POS                    <QJ>
9638 ENDIF                               <UP>
9790 ENDPROC                             <PZ>
9792 --                                  <XJ>
9793 REM -  Player Missiles nach  -      <OG>
9794 REM -  rechts bewegen       -       <GM>
9799 --                                  <YL>
9800 PROC PMRIGHT                        <ZV>
9820 IF PMNR=0                           <VW>
9822 X0POS=X0POS+1                       <TL>
9824 IF X0POS>199 THEN X0POS=199         <XE>
9826 POKE 53248,X0POS                    <PD>
9828 ENDIF                               <UQ>
9830 IF PMNR=1                           <WN>
9832 X1POS=X1POS+1                       <UI>
9834 IF X1POS>198 THEN X1POS=198         <WT>
9836 POKE 53249,X1POS                    <QN>
9838 ENDIF                               <UT>
9990 ENDPROC                             <QD>
10000 --                                 <JC>
10002 REM -  Variablen und Player  -     <FN>
10003 REM -  Missiles definieren   -     <KZ>
10004 --                                 <JW>
10009 PROC DECLARE                       <VM>
10010 DIM FILE$(20),DAT$(20),SCREEN$(768
0),LEER$(256),LO$(10),RU$(10),E$(20)     <WZ>
10020 COL0=40:COL1=202:COL2=148:COL3=0   <AJ>
10030 LEER$(1)="{CTL ,}":LEER$(256)="{CT
L ,}":LEER$(2)=LEER$                     <IW>
10040 LO$="{CTL ,}{CTL O}{CTL H}{CTL H}{
CTL H}{CTL H}{CTL H}{CTL ,}":RU$="{CTL ,
}{CTL A}{CTL A}{CTL A}{CTL A}{CTL A}{CTL
 O}{CTL ,}"                              <EZ>
10100 ENDPROC                            <UA>
11000 --                                 <JE>
11002 REM -  Apfelmaennchen auf   -      <GF>
11003 REM -  dem Bildschirm dar-  -      <CJ>
11004 REM -  stellen              -      <UC>
11005 --                                 <KD>
11010 PROC BILDHOLEN                     <IX>
11015 GRAPHICS 15+16                     <YR>
11020 MOVE ADR(SCREEN$),DPEEK(88),7680   <IF>
11030 POKE 708,COL0:POKE 709,COL1:POKE 7
10,COL2:POKE 712,COL3                    <IB>
11040 ENDPROC                            <UP>
11100 --                                 <JH>
11102 REM -  Bild speichern und   -      <HU>
11103 REM -  Parameter anzeigen   -      <UN>
11104 --                                 <KB>
11110 PROC BILDSPEICHERN                 <HD>
11120 MOVE DPEEK(88),ADR(SCREEN$),7680   <AG>
11130 ENDPROC                            <UO>
15000 PROC PARAMANZ                      <PR>
15010 CLS                                <OD>
15020 POSITION 9,3:? "PARAMETER▲ANZEIGEN
"                                        <TP>
15030 POSITION 9,4:? "------------------
"                                        <CC>
15040 POSITION 3,7:? "Linker▲Rand▲=▲";XM
IN                                       <OZ>
15050 POSITION 3,9:? "Rechter▲Rand▲=▲";X
MAX                                      <RQ>
15060 POSITION 3,11:? "Unterer▲Rand▲=▲";
YMAX                                     <OJ>
15070 POSITION 3,13:? "Oberer▲Rand▲=▲";Y
MIN                                      <HX>
15080 POSITION 3,15:? "Tiefe▲=▲";TMAX    <JY>
15100 GET EIN                            <LG>
15110 ENDPROC                            <UO>
15190 --                                 <KZ>
15200 REM -  Fehlerbehandlung      -     <ZG>
15210 --                                 <JW>
20000 REM FEHLERROUTINE                  <ND>
20010 ? "{ESC CTL 2}"                    <DP>
20015 POSITION 3,10:? "FEHLER▲NR.▲";ERR;
"▲IST▲AUFGETRETEN"                       <OH>
20017 POSITION 3,12:? "ZEILEN▲NR.▲";ERL  <ZA>
20020 IF ERR=170 THEN POSITION 3,20:? "D
ATEI▲NICHT▲VORHANDEN"                    <SZ>
20030 IF ERR=162 THEN POSITION 3,20:? "D
ISKETTE▲IST▲VOLL"                        <EJ>
20040 IF ERR=164 THEN POSITION 3,20:? "D
ATEI▲NICHT▲IN▲ORDNUND"                   <IG>
20050 IF ERR=169 THEN POSITION 3,20:? "D
IRECTORY▲IST▲VOLL"                       <QF>
20060 IF ERR=160 THEN POSITION 3,20:? "D
ISKETTENLAUFWERK▲NICHT▲BEKANNT"          <SV>
20070 IF ERR=128 THEN POSITION 3,20:? "U
EBERTRAGUNG▲UNTERBROCHEN"                <VF>
20080 IF ERR=167 THEN POSITION 3,20:? "D
ATEI▲IST▲SCHREIBGESCHUETZT"              <ZX>
20090 IF ERR=144 THEN POSITION 3,20:? "D
ISKETTE▲IST▲SCHREIBGESCHUETZT"           <TI>
20100 IF ERR=138 THEN POSITION 3,20:? "G
ERAET▲NICHT▲ANSPRECHBAR"                 <JT>
20110 IF ERR=165 THEN POSITION 3,20:? "F
ALSCHER▲DATEINAME"                       <AB>
20115 POSITION 3,22:? "WEITER▲MIT▲RETURN
▲!"                                      <WP>
20120 GET EIN:TRAP 40000                 <AU>
20130 GOTO 95                            <WZ>
```

Turbo-Basic XL-Listing zu »Apfelmännchen« (Schluß)







Grafikspielereien in Turbo-Basic XL

Mit einfachen Mitteln lassen sich unter Turbo-Basic XL schöne Grafikeffekte erzeugen.

Die neuen Befehle von Turbo-Basic XL eröffnen dem Programmierer eine Reihe zusätzlicher Möglichkeiten, die unter dem normalen Atari-Basic nur mit Hilfe von Assembler-Unterroutinen realisierbar waren. So erlaubt auch der Turbo-Basic-Befehl »MOVE«, der Speicherbereiche blitzschnell verschieben kann, für Basic völlig neue Programmtechniken. Anhand eines Grafikbildes zeigt das Programm »Turbo-Basic-Grafikdemo«, wie sich die Daten des Bildschirmspeichers schnell und einfach verschieben oder speichern lassen.

Das Programm ist in einzelne Prozeduren unterteilt, die am Anfang aufgerufen werden. Mit Ausnahme der Prozeduren »DECLARE« und »DEMO« spielt die Reihenfolge des Aufrufs keine Rolle. Die Routinen, die an der Grafik Veränderungen vornehmen, sind in sich abgeschlossen und stellen vor ihrem Ende stets den ursprünglichen Zustand des Bildes her. Die Prozeduren bewirken im einzelnen folgendes:

Prozedur »DECLARE«

Initialisieren und dimensionieren der benötigten Stringvariablen.

Prozedur »DEMO«

Erzeugen einer Spirale im Grafik 15-Format. An diesem Bild werden alle weiteren Manipulationen vorgenommen. Hier kann man natürlich sowohl andere Grafiken erzeugen als auch ein Bild von Diskette laden.

Prozedur »TEXTEIN«

Einblenden eines Textes in das Grafikbild. Der dabei überschriebene Teil der Grafik wird zuvor in einem String gespeichert. So ist sichergestellt, daß man den ursprünglichen Zustand des Bildes wiederherstellen kann.

Prozedur »XMIRROR«

Spiegeln der Grafik um die X-Achse. Das Bild wird also auf den Kopf gestellt.

Prozedur »XKLAPPT«

Spiegeln der oberen Hälfte des Bildschirms nach unten.

Prozedur »XKLAPPB«

Hierbei handelt es sich im Prinzip um die gleiche Routine wie XKLAPPT. Es wird lediglich die untere Hälfte nach oben gespiegelt.

Prozedur »XCOPYT«

Kopieren der oberen Bildschirmhälfte nach unten.

Prozedur »XCOPYB«

Untere Hälfte des Bildes nach oben kopieren.

Prozedur »YCOPYL«

Die linke Hälfte des Bildschirms nach rechts kopieren.

Prozedur »YCOPYR«

Die rechte Bildschirmhälfte nach links kopieren.

Prozedur »PUTPIC«

Ablegen des Bildschirminhalts in einem String.

Prozedur »GETPIC«

Der Inhalt eines Strings wird auf den Bildschirm geschrieben.

Prozedur »EFFEKTE«

Die Inhalte der Farbregister 708, 709 und 710 werden ausgetauscht. Dadurch entsteht ein schöner Farbeffekt.

Alle Werte in den beschriebenen Prozeduren beziehen sich übrigens auf ein Bild in Grafikstufe 15. Selbstverständlich können die Effekte auch in allen anderen Grafikstufen verwendet werden. Man muß dann lediglich die Größe des Bildschirmspeichers sowie die Anzahl der benötigten Byte pro Zeile entsprechend anpassen. In Grafikstufe 15 errechnet sich die Bildschirmgröße beispielsweise aus 192 Zeilen und 40 Byte pro Zeile. 192 multipliziert mit 40 ergibt also 7680.

(Wolfgang Czerny/wb)

PROGRAMM-STECKBRIEF	
Programmname	Grafik-Spielereien
Programmtyp	Grafik
Programmiersprache	Turbo-Basic XL
Programmlänge	7939 Byte
für Computer	Atari 800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	Prüfsummer
Bemerkung	Effektvolle Grafikmanipulation
Leserservice	Diskette SPIELE.TUR

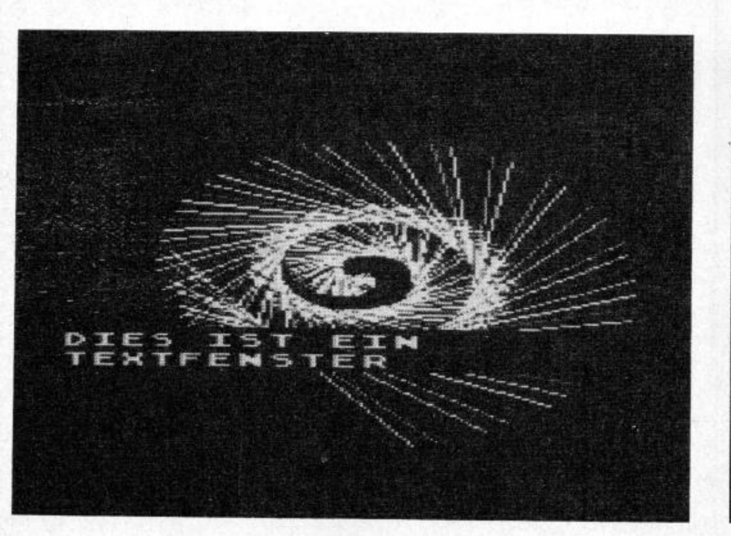


Bild 1. Ein Grafik 8-Bild mit Textfenster

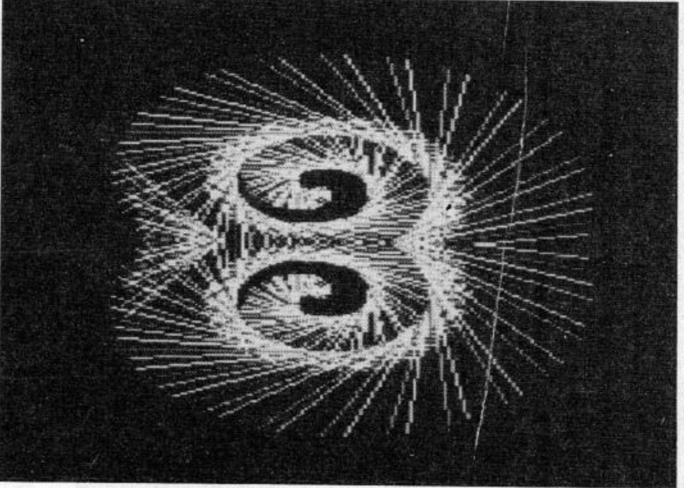


Bild 2. Hier ist das Textfenster wieder gelöscht und die Grafik von Bild 1 gespiegelt

```
100 REM SPIELEREIEN MIT DEM                <WT>
110 REM MOVE BEFEHL IN TURBO BASIC         <ZD>
115 REM ------------                       <QD>
120 REM VON WOLFGANG CZERNY                <XN>
130 --                                     <JI>
160 EXEC DECLARE                           <LK>
170 EXEC DEMO                              <WV>
175 DO                                     <FC>
177 EXEC TEXTEIN                           <IQ>
180 EXEC XKLAPPT                           <KP>
190 EXEC XKLAPPB                           <ZP>
200 EXEC XMIRROR                           <RG>
210 EXEC XCOPYT                            <XB>
220 EXEC XCOPYB                            <MT>
230 EXEC YCOPYL                            <SZ>
240 EXEC YCOPYR                            <WN>
250 LOOP                                   <CH>
500 END                                    <XQ>
999 --                                     <LD>
1000 PROC DECLARE                          <DD>
1010 DIM A$(8000),B$(8000),LEER$(800)      <NC>
1020 A$(1)="{CTL ,}":A$(8000)="{CTL ,}":
A$(2)=A$                                   <HV>
1025 B$(1)="{CTL ,}":B$(8000)="{CTL ,}":
B$(2)=B$                                   <LX>
1030 LEER$(1)="{CTL ,}":LEER$(800)="{CTL
,}":LEER$(2)=LEER$                         <JV>
1040 ENDPROC                               <OO>
1090 --                                    <WF>
2000 PROC TEXTEIN                          <BK>
2010 EXEC PUTPIC                           <YR>
2020 MOVE DPEEK(88)+3840,ADR(B$),800       <LS>
2030 MOVE ADR(LEER$),DPEEK(88)+3840,800    <LF>
2040 TEXT 0,98,"DIES IST EIN"              <IN>
2050 TEXT 0,107,"TEXTFENSTER"              <JU>
2060 PAUSE 400                             <WE>
2070 MOVE ADR(B$),DPEEK(88)+3840,800       <SF>
2080 PAUSE 400                             <WK>
2090 EXEC GETPIC                           <OK>
2100 ENDPROC                               <OF>
2110 --                                    <VK>
5000 PROC XMIRROR                          <MD>
5010 FOR J=1 TO 10                         <DI>
5020 EXEC PUTPIC                           <YX>
5030 FOR I=7680 TO 0 STEP -40              <PX>
5040 MOVE ADR(A$)+I-40,DPEEK(88)+7680-I,
40                                         <LG>
5050 NEXT I                                <FI>
5060 NEXT J                                <FW>
5070 PAUSE 500                             <WW>
5075 EXEC GETPIC                           <PB>
5080 ENDPROC                               <PE>
5090 --                                    <WJ>
6000 PROC XKLAPPT                          <EJ>
6020 EXEC PUTPIC                           <YY>
6030 FOR I=3840 TO 0 STEP -40              <MY>
6040 MOVE ADR(A$)+I-40,DPEEK(88)+7680-I,
40                                         <LH>
6050 NEXT I                                <FJ>
6070 EXEC EFFEKTE                          <FT>
6080 EXEC GETPIC                           <OL>
6090 ENDPROC                               <PI>
6099 --                                    <XU>
7000 PROC XKLAPPB                          <SQ>
7020 EXEC PUTPIC                           <YZ>
7030 FOR I=7680 TO 3840 STEP -40           <CO>
7040 MOVE ADR(A$)+I-40,DPEEK(88)+7680-I,
40                                         <LI>
7050 NEXT I                                <FK>
7070 EXEC EFFEKTE                          <FU>
7080 EXEC GETPIC                           <OM>
7090 ENDPROC                               <PJ>
7099 --                                    <XV>
8000 PROC XCOPYT                           <PB>
8020 EXEC PUTPIC                           <ZA>
8030 FOR I=0 TO 3840 STEP 40               <NH>
8040 MOVE ADR(A$)+I,DPEEK(88)+3800+I,40    <KF>
8050 NEXT I                                <FL>
8060 PAUSE 500                             <WW>
8070 EXEC GETPIC                           <OK>
8080 ENDPROC                               <PH>
8090 --                                    <WM>
9000 PROC XCOPYB                           <EA>
9020 EXEC PUTPIC                           <ZB>
9030 FOR I=3840 TO 7680 STEP 40            <AS>
9040 MOVE ADR(A$)+I,DPEEK(88)-3840+I,40    <RQ>
9050 NEXT I                                <FM>
9060 PAUSE 500                             <WX>
9070 EXEC GETPIC                           <OL>
9080 ENDPROC                               <PI>
9999 --                                    <YP>
10000 PROC YCOPYL                          <AH>
10020 EXEC PUTPIC                          <NV>
10030 FOR I=0 TO 191                       <IZ>
10040 MOVE DPEEK(88)+I*40,DPEEK(88)+I*40
+20,20                                     <UJ>
10050 NEXT I                               <GD>
10060 PAUSE 300                            <BY>
10070 EXEC GETPIC                          <CL>
10080 ENDPROC                              <VD>
10099 --                                   <MF>
10100 PROC YCOPYR                          <EI>
10120 EXEC PUTPIC                          <NY>
10130 FOR I=0 TO 191                       <JC>
10140 MOVE DPEEK(88)+I*40+20,DPEEK(88)+I
*40,20                                     <LX>
10150 NEXT I                               <GG>
10160 PAUSE 300                            <CB>
10170 EXEC GETPIC                          <CO>
10180 ENDPROC                              <VG>
10199 --                                   <MI>
11000 PROC GETPIC                          <GA>
11010 MOVE ADR(A$),DPEEK(88),7680          <SR>
11020 ENDPROC                              <UH>
11999 --                                   <NI>
12000 PROC PUTPIC                          <SG>
12010 MOVE DPEEK(88),ADR(A$),7680          <EJ>
12020 ENDPROC                              <UJ>
13000 PROC EFFEKTE                         <ZZ>
13010 POKE 764,255                         <RH>
13020 REPEAT                               <HM>
13030 POKE 708,40:POKE 709,202             <ZJ>
13040 POKE 710,148:PAUSE 5                 <VD>
13050 POKE 708,202:POKE 709,148            <UJ>
13060 POKE 710,40:PAUSE 5                  <NS>
13070 POKE 708,148:POKE 709,40             <KU>
13080 POKE 710,202:PAUSE 5                 <PP>
13090 UNTIL PEEK(764)<>255                 <DJ>
13100 ENDPROC                              <UG>
13110 --                                   <JP>
15000 PROC DEMO                            <PW>
15010 XI=80:YI=70:Q=0:DEG                  <IA>
15020 GRAPHICS 31:C=20                     <CS>
15040 FOR I=1 TO 1000 STEP 5               <OD>
15050 Q=Q+1:IF Q>3.5 THEN Q=1              <JL>
15060 COLOR Q:R=I/10:T=I                   <FU>
15070 X=R*COS(T):Y=R*SIN(T)                <CC>
15080 IF Y+YI>151 THEN 15140               <BE>
15090 PLOT X+XI,Y+YI                       <WW>
15100 X=(I+C)/16*COS(I+C+90)               <AE>
15110 Y=(I+C)/16*SIN(I+C+90)               <DL>
15120 DRAWTO X+XI,Y+YI                     <JT>
15130 NEXT I                               <GI>
15140 ENDPROC                              <VA>
```

Listing zu »Turbo-Basic-Grafikdemo«

Mehr Tempo für Player Missiles

In normalem Basic ist die Positionierung von Playern und Missiles eine zeitaufwendige Angelegenheit, in Turbo-Basic XL dagegen nicht.

Wer sich schon einmal mit der Player-Missile-Grafik der Atari-Computer auseinandergesetzt hat, der weiß, daß die horizontale Verschiebung recht einfach ist. Die vertikale Positionierung hingegen gestaltet sich schon wesentlich aufwendiger. Das Programm »PMMOVE« demonstriert, wie man den MOVE-Befehl in Verbindung mit der Player-Missile-Grafik sinnvoll einsetzen kann. Die Auswahl, welchen Player man bewegen möchte, erfolgt mit den Tasten 1 bis 4. Und hier eine kurze Programmbeschreibung:

Prozedur »DECLARE«

Hier werden die Stringvariablen dimensioniert und vier Strings mit den Werten für die verschiedenen Player versehen. Die Höhe jedes Players beträgt in diesem Fall acht Zeilen. Ober- und unterhalb ist jeweils eine Leerzeile hinzugefügt, damit beim Verschieben der alte Player nicht jedesmal extra gelöscht werden muß.

Prozedur »INIT«

Hier werden die Basiswerte für die Player-Missile-Erzeugung festgelegt. Zudem wird der Speicherbereich der Player gelöscht. Auch diese Maßnahme wird mit Hilfe des MOVE-Befehls durchgeführt. Die Anfangspositionen der vier Player sind durch die Felder X() und Y() festgelegt.

Prozedur »PM-BAUEN«

Die Strings, die für das Aussehen der Player verantwortlich sind, werden durch den MOVE-Befehl in die entsprechenden Speicherbereiche verlegt. Es folgt dann die Fixierung der horizontalen Positionen.

(Wolfgang Czerny/wb)

PROGRAMM-STECKBRIEF	
Programmname	Player/Missile-Mover
Programmtyp	Utility
Programmiersprache	Turbo-Basic XL
Programmlänge	2372 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	keine
Eingabehilfe	AMPEL
Bemerkung	Das Programm zeigt, wie man mit den speziellen Turbo-Basic-Befehlen Player Missiles behandeln kann
Leserservice	Diskette (PMMOVE.TUR)

```
100 REM PLAYER/MISSLE STEUERUNG                  <RV>
110 REM .   IN TURBO-BASIC                       <KB>
120 REM                                          <AM>
130 REM von Wolfgang Czerny                      <WV>
140 REM                                          <AQ>
145 --                                           <JZ>
150 EXEC DECLARE                                 <LI>
155 GRAPHICS 31                                  <XC>
160 EXEC INIT                                    <DD>
170 EXEC PM_BAUEN                                <CD>
1000 DO                                          <TM>
1010 REPEAT                                      <EC>
1020 GET EIN                                     <IF>
1030 UNTIL (EIN>48 AND EIN<53) OR EIN=42
 OR EIN=43 OR EIN=45 OR EIN=61                   <DI>
1040 IF EIN>48 AND EIN<53 THEN NR=EIN-48 <QA>
1050 IF EIN=42 THEN X(NR)=X(NR)+1                <NO>
1060 IF EIN=43 THEN X(NR)=X(NR)-1                <QR>
1070 IF EIN=45 THEN Y(NR)=Y(NR)-1                <TS>
1080 IF EIN=61 THEN Y(NR)=Y(NR)+1                <QF>
1090 IF Y(NR)>255 THEN Y(NR)=0                   <MP>
1100 IF Y(NR)<0 THEN Y(NR)=255                   <CO>
1110 IF X(NR)>255 THEN X(NR)=0                   <KM>
1120 IF X(NR)<0 THEN X(NR)=255                   <BP>
1130 POKE 53247+NR,X(NR)                         <VO>
1140 IF NR=1 THEN MOVE ADR(LOOK1$),PMB+1
024+Y(1),10                                      <HX>
1150 IF NR=2 THEN MOVE ADR(LOOK2$),PMB+1
280+Y(2),10                                      <RT>
1160 IF NR=3 THEN MOVE ADR(LOOK3$),PMB+1
536+Y(3),10                                      <BY>
1170 IF NR=4 THEN MOVE ADR(LOOK4$),PMB+1
792+Y(4),10                                      <LU>
2000 LOOP                                        <XV>
2010 END                                         <PT>
2099 --                                          <XQ>
10000 PROC DECLARE                               <TT>
10010 DIM Y(4),X(4)                              <AW>
10030 DIM LOOK1$(10),LOOK2$(10)                  <FC>
10040 DIM LOOK3$(10),LOOK4$(10)                  <IK>
10050 DIM LEER$(256)                             <CB>
10060 POKE 730,1                                 <UF>
10070 RESTORE                                    <DO>
10080 FOR I=1 TO 10                              <RZ>
10090 READ W:LOOK1$(I,I)=CHR$(W)                 <SE>
10100 NEXT I                                     <FM>
10110 FOR I=1 TO 10                              <RA>
10120 READ W:LOOK2$(I,I)=CHR$(W)                 <RX>
10130 NEXT I                                     <FY>
10140 FOR I=1 TO 10                              <RM>
10150 READ W:LOOK3$(I,I)=CHR$(W)                 <TB>
10160 NEXT I                                     <GK>
10170 FOR I=1 TO 10                              <RY>
10180 READ W:LOOK4$(I,I)=CHR$(W)                 <UF>
10190 NEXT I                                     <GW>
10500 REM LOOK1                                  <IT>
10510 DATA.0,1,2,4,8,144,224,224,240,0           <WO>
10520 REM LOOK2                                  <JQ>
10530 DATA.0,240,224,224,144,8,4,2,1,0           <RG>
10540 REM LOOK3                                  <KN>
10550 DATA.0,128,64,32,16,9,7,7,15,0             <ZL>
10560 REM LOOK4                                  <LK>
10570 DATA.0,15,7,7,9,16,32,64,128,0             <BV>
10990 ENDPROC                                    <WI>
10999 --                                         <NG>
11000 PROC INIT                                  <XC>
11010 PMB=PEEK(106)-40:POKE 54279,PMB            <RL>
11020 PMB=PMB*256:NR=1                           <BU>
11030 POKE 559,62:REM SINGLE RES.                <YC>
11040 FOR I=53256 TO 53259                       <PR>
11050 POKE I,0                                   <VK>
11060 NEXT I                                     <GJ>
11070 FOR I=704 TO 707                           <WV>
11080 POKE I,255                                 <ID>
11090 NEXT I                                     <GV>
11100 FOR I=1024 TO 1792 STEP 256                <GJ>
11110 MOVE ADR(LEER$),PMB+I,256                  <OA>
11120 NEXT I                                     <FW>
11140 X(1)=50:Y(1)=50:X(2)=60                    <OQ>
11150 Y(2)=60:X(3)=70:Y(3)=70                    <VB>
11160 X(4)=80:Y(4)=80                            <JR>
11190 ENDPROC                                    <VM>
11199 --                                         <MK>
12000 PROC PM_BAUEN                              <TS>
12010 MOVE ADR(LOOK1$),PMB+1024+Y(1),10  <XO>
12020 MOVE ADR(LOOK2$),PMB+1280+Y(2),10  <EF>
12030 MOVE ADR(LOOK3$),PMB+1535+Y(3),10  <KA>
12040 MOVE ADR(LOOK4$),PMB+1792+Y(4),10  <RW>
12050 FOR I=1 TO 4                               <KR>
12060 POKE 53247+I,X(I)                          <ZR>
12070 NEXT I                                     <GP>
12080 POKE 53277,3                               <QY>
12090 ENDPROC                                    <VL>
```

Programm zur Player-Missile-Steuerung (Turbo-Basic XL)

Daten komprimiert gespeichert

Wer Speicherplatz auf Diskette sparen möchte, der kann in Turbo-Basic XL zum Speichern und Laden von numerischen Daten auch die Befehle »%PUT« und »%GET« verwenden.

Das Atari-Basic kennt zum Speichern und Laden von Daten mehrere Anweisungen. Zum Schreiben benutzt man die Befehle PRINT und PUT, zum Lesen die Befehle INPUT und GET. Unter Turbo-Basic XL allerdings verfügt man neben den Anweisungen BPUT und BGET, die für die Blockspeicherung zuständig sind, auch über die Befehle %PUT und %GET.

In diesem Fall sollen jedoch nur diejenigen Befehle betrachtet werden, mit denen man Gleitkommazahlen auf Diskette speichert oder von Diskette lädt. Wie das Listing zeigt, kann man dazu die Befehle PRINT und INPUT beziehungsweise %PUT und %GET verwenden. Diese beiden Befehlspaare führen zwar dieselbe Operation aus, gehen dabei aber unterschiedliche Wege. Speichert man eine Gleitkommazahl mit Hilfe des PRINT-Befehls, so benötigt jede Zahl 8 Byte Speicherplatz auf Diskette. Verwendet man hingegen für denselben Vorgang den %PUT-Befehl, benötigt eine Gleitkommazahl nur mehr 6 Byte Speicherplatz.

Im Listing werden nun 1 000 Zufallszahlen in einem Feld abgelegt und auf die beiden verschiedenen Methoden auf Diskette gespeichert. Anschließend laden wir sie wieder. Wie man der Tabelle entnehmen kann, spart man bei der Speicherung von nur 1 000 Zahlen mit den %PUT- und %GET-Befehlen immerhin 56 Sektoren auf Diskette. Das sind rund 6,8 KByte. Es versteht sich, daß gespeicherte Zahlen jeweils nur von dem äquivalenten Lese-Befehl gelesen werden können.

Neben dem geringeren Speicherbedarf hat die Verwendung von %PUT und %GET auch noch einen weiteren Vorteil. Die benötigten Schreib- und Lesezeiten verringern sich um etwa 50 Prozent. Die genauen Zeiten sind in der Tabelle festgehalten. Hierbei wurden übrigens drei verschiedene Disketten-Laufwerke verwendet. Ein Atari-810-Laufwerk, das neuere 1050-Laufwerk und ein aufgerüstetes 1050-Laufwerk (wie im Artikel »Rasende Daten« in diesem Heft beschrieben).

(Wolfgang Czerny/wb)

PROGRAMM-STECKBRIEF	
Programmname	%PUT- und %GET-Demo
Programmtyp	Demo
Programmiersprache	Turbo-Basic XL
Programmlänge	1008 Byte
für Computer	alle
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	Prüfsummer
Bemerkung	
Leserservice	Diskette (PUTGET.TUR)

Basic-Befehl	Ausführungszeiten in Sekunden						Benötigte Sektoren
	Erweitertes 1050-Laufwerk		Normales 1050-Laufwerk		Atari-810-Laufwerk		
	Mit Verify	Ohne Verify	Mit Verify	Ohne Verify	Mit Verify	Ohne Verify	
PRINT	32.74	27.48	58.22	35.32	56.22	34.02	104
INPUT	21.2	20.68	34.48	34.34	32.48	32.48	
%PUT	19.34	16.68	28.98	17.64	27.30	16.50	48
%GET	11.42	11.86	16.28	16.12	15.26	15.26	

Tabelle mit den Ausführungszeiten unter Verwendung der verschiedenen Befehle, mit denen sich Daten speichern und laden lassen.

```
10 --                                              <YW>
20 REM -     %PUT und %GET Demo      -             <AR>
30 REM -     von Frank Ostrowski     -             <YF>
40 --                                              <YZ>
100 ? "%PUT▲Demonstration"                         <ZZ>
110 DIM A(1000)                                    <WX>
120 ? "Creating▲Array"                             <DB>
130 FOR I=1 TO 1000                                <GZ>
140 A(I)=RND                                       <FK>
150 NEXT I                                         <GB>
160 --                                             <JO>
170 ? "PRINT▲File"                                 <AK>
180 --                                             <JS>
190 T=TIME                                         <PG>
200 OPEN #1,8,0,"D:RNDTST.DAT"                     <SX>
210 FOR I=1 TO 1000                                <GW>
220 ? #1;A(I)                                      <UU>
230 NEXT I                                         <FY>
240 CLOSE #1                                       <LG>
250 ? "▲PRINT:▲";(TIME-T)/50;"▲Sekunden"           <DC>
260 --                                             <JP>
270 T=TIME                                         <PD>
280 OPEN #1,4,0,"D:RNDTST.DAT"                     <RN>
290 FOR I=1 TO 1000                                <HM>
300 INPUT #1,A:A(I)=A                              <RF>
310 NEXT I                                         <FV>
320 CLOSE #1                                       <LD>
330 ? "▲INPUT:▲";(TIME-T)/50;"▲Sekunden"           <FF>
340 DIR "D:RNDTST.DAT"                             <MV>
350 DELETE "D:RNDTST.DAT"                          <QX>
360 --                                             <JQ>
370 ? "▲%PUT▲File"                                 <BU>
380 --                                             <JU>
390 T=TIME                                         <PI>
400 OPEN #1,8,0,"D:RNDTST.DAT"                     <SZ>
410 FOR I=1 TO 1000                                <GY>
420 %PUT #1;A(I)                                   <YN>
430 NEXT I                                         <GA>
440 CLOSE #1                                       <LI>
450 ? "▲%PUT▲:▲";(TIME-T)/50;"▲Sekunden"           <VJ>
460 --                                             <JR>
470 T=TIME                                         <PF>
480 OPEN #1,4,0,"D:RNDTST.DAT"                     <RP>
490 FOR I=1 TO 1000                                <HO>
500 %GET #1,A:A(I)=A                               <EG>
510 NEXT I                                         <FX>
520 CLOSE #1                                       <LF>
530 ? "▲%GET▲:▲";(TIME-T)/50;"▲Sekunden"           <LA>
540 DIR "D:RNDTST.DAT"                             <MX>
550 DELETE "D:RNDTST.DAT"                          <QZ>
```

Listing zu »%PUT und %GET« (Turbo-Basic XL)

Ordnung muß sein

»Wer suchet, der findet« ist das Motto vieler Programmsammlungen. Mit »Happy-Disksorter« wird die Verwaltung einer Softwaresammlung zum Kinderspiel. Sehen Sie selbst.

Eine Programmsammlung kann so schnell wachsen, daß man den Überblick über seine wertvolle Software verliert. Dann legt man Karteikarten an, oder hält auf losen Zetteln fest, welche Programme sich auf welcher Diskette befinden. Nun, ideal scheinen diese Lösungen nicht. Da sich ein Computer aber auch hervorragend zur Datenspeicherung eignet, liegt es nahe, ihm diese Aufgabe der Programmverwaltung zu übertragen. Dazu benötigt man natürlich spezielle Software.

»Happy-Disksorter« besteht aus zwei Teilen. Erstens dem Maschinenspracheteil (Listing 1) mit dem Menübild und der geänderten Display-List, zweitens dem Hauptprogramm (Listing 2). Gearbeitet wird nur mit dem Hauptprogramm. Listing 1 wird stets hinzugeladen und liegt dann sozusagen im Hintergrund vor.

Bevor Sie mit »Happy-Disksorter« arbeiten können, müssen beide Listings eingetippt sein. Dann laden und starten Sie zuerst Listing 1. Dieses Programm erzeugt das Maschinenfile mit dem Namen »MENU.PIC« auf Diskette. Anschließend brauchen Sie nur noch Listing 2 aufzurufen und zu starten. Es folgt dann automatisch das File »MENU.PIC« und Sie gelangen in das Hauptmenü vom »Happy-Disksorter« (Bild). Beachten Sie, daß sich dieses File unbedingt auf der gleichen Diskette wie das Hauptprogramm befindet. Später brauchen Sie dann nur noch das Hauptprogramm aufzurufen, da das Maschinensprache-File nur einmal auf Diskette geschrieben werden muß.

Befinden Sie sich einmal im Hauptmenü, können Sie schon die ersten Programmnamen eingeben. Die Namen dürfen maximal 25 Zeichen lang sein. Anschließend folgt die Frage nach dem Umfang des Programms in Sektoren und des Indexes zum schnellen Auffinden von Programmen.

Um ins Menü des »Happy-Disksorters« zu gelangen, betätigt man bei der Frage nach dem Programmnamen einfach die RETURN-Taste. Daraufhin erscheint am linken oberen Bildschirmrand ein Pfeil, den man mit den Cursortasten zu dem gewünschten Symbol bewegen kann. Die nochmalige Betätigung der RETURN-Taste führt die Funktion aus.

Suchen

Man gibt entweder die Eintragsnummer, mit der jedes Programm versehen wird, oder einen unverwechselbaren Teil des gesuchten Programmnamens ein. Sollte ein Eintrag mit dem Suchbegriff übereinstimmen, wird er auf dem Bildschirm ausgegeben. Durch Drücken der RETURN-Taste gelangt man wieder zum Ausgangsmenü zurück.

Listen

Die eingetragenen Namen in abfallender Reihenfolge auf dem Bildschirm ausgeben. Ist der Bildschirm mit Namen gefüllt, wird nach Betätigung der Leertaste die nächste Seite auf dem Bildschirm ausgegeben. Mit der ESC-Taste gelangt man jederzeit zurück zum Hauptmenü. Die Ausgabe eines bestimmten Eintrags bewirkt die RETURN-Taste.

Sortieren

Nach dem Index oder dem Namen sortieren.

Eintrag löschen

Die Auswahl der zu löschenden Daten erfolgt wie unter »Suchen«. Der gesuchte Programmname erscheint dann auf dem Bildschirm und wird durch Drücken der Taste J gelöscht.

Disketten-Menü

Mit diesem Symbol gelangt man in ein weiteres Menü:
1. Auflisten des Directorys in Laufwerk 1. Der Listvorgang läßt sich mit der START-Taste oder CONTROL-1 unterbrechen. Am Ende gelangt man mit einer beliebigen Taste zurück ins Untermenü.
2. Speichern der im RAM-Speicher vorliegenden Programmnamen. Es genügt zur Erkennung des Dateinamens die ersten acht Zeichen einzugeben. Die RETURN-Taste bringt den Benutzer wieder zurück ins Untermenü.
3. Laden einer auf Diskette gespeicherten Programmliste. Die Eingabe erfolgt wie unter Punkt 2 beschrieben.
4. Alle Eintragungen einer zu katalogisierenden Diskette werden automatisch in die im RAM-Speicher vorliegende Datei übernommen. Durch Drücken irgendeiner Taste, ausgenommen 1 bis 4, gelangt man zum Hauptmenü zurück.

Liste drucken

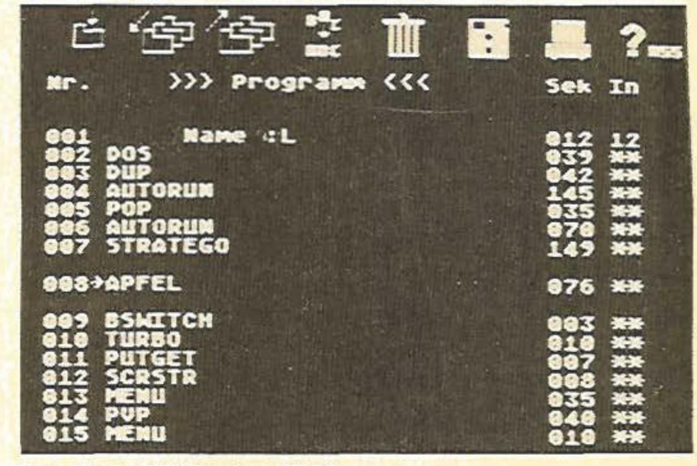
Es besteht die Wahl zwischen einem zwei- und dreispaltigem Ausdruck. Als Drucker ist ein Epson FX80 vordefiniert. Auch hier gelangt man mit Betätigung irgend einer Taste zum Hauptmenü zurück.

Fehlermeldungen«

Sollte ein Fehler auftreten, springt das Programm automatisch ins Hauptmenü zurück. Wählt man anschließend die Funktion »Fehlermeldungen«, wird der Fehler auf dem Bildschirm beschrieben. (Andreas Kubelka/wb)

Zeile 7040 – Die Zeichen, die zwischen Anführungszeichen stehen, bewirken einen Ausdruck im normalen Modus. Diese Einstellung ist für zweispaltigen Ausdruck mit 80 Zeichen pro Zeile verantwortlich.
Zeile 7100 – Umschaltung auf Ausdruck mit 132 Zeichen pro Zeile und in drei Spalten.
Es müssen also stets die Zeichen zwischen Anführungszeichen geändert werden. Am besten orientieren Sie sich an Ihrem Druckerhandbuch.

PROGRAMM-STECKBRIEF	
Programmname	Happy-Disksorter
Programmtyp	Utility
Programmiersprache	Atari-Basic
Programmlänge	10034 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	Prüfsummer
Bemerkung	Das Programm besteht aus zwei Teilen. Zuerst Listing 1 starten und anschließend Listing 2.
Leserservice	Diskette (DISKSORT.BAS/ DISKMENU.BAS)



Das »Happy-Disksorter«-Menü

Zeile	Beschreibung
10 bis 200	Initialisierung
300 bis 390	Eingabe des Namens mit Anzahl der Sektoren und dem Index
400 bis 440	Menü-Auswahl
1000 bis 1050	Auflisten der Namen
1500 bis 1510	Suchen eines Namens aus der Liste
2000 bis 2110	Name in der gesamten Liste suchen
2500 bis 2510	Eingabe des gesuchten Namens
3000	Parameter setzen für Listen in abfallender Reihenfolge
4000 bis 4040	Sortierroutine
5000 bis 5200	Name löschen
6000 bis 6050	Disketten-Untermenü
6100 bis 6140	Schreiben des Directorys
6200	Eingabe des Filenamens
6410 bis 6440	Liste abspeichern/laden
6500 bis 6530	Übernahme des Directorys in die Liste
7000 bis 7130	Druckerausgabe
7400	Name in Kanal D: schreiben
7500 bis 7510	Namennummer in Kanal D: schreiben
7600	Überschrift in Kanal D: schreiben
8000 bis 9200	Fehlermeldung

Programmbeschreibung

Variable	Beschreibung
US	—enthält die Adresse der Maschinenroutine zum Laden und Speichern
D$	—String für alle Daten
A$	—Nimmt Eingaben entgegen
P	—Position des Pfeils beim Auflisten
C	—Zähler
D	—Enthält Kanalnummer, in den geschrieben wird Kanal : 0 = Bildschirm Kanal : 1 = Drucker
ER	—Fehlercode
LD	—Enthält die zu bearbeitende Position in D$
L	—Länge von A$
I	—Für Schleifen und Pfeilposition beim Menü
A	—Nimmt Eingaben über GET entgegen
CNT	—Anzahl der Daten in D$
STA	—Die Startnummer beim Listen oder die aktuelle Position
STO	—Stopposition beim Listen
PLMI	—Gibt die Listrichtung an
F	—Gibt beim Suchen die Position an, an der die Eingabe gefunden wurde
R	—Restwert für die Druckerroutine, um die Startnummer der nächsten Spalte zu finden.
S2	—Startnummer der 2. Spalte beim Drucken von 3 Spalten

Variablenliste

Einsprungadressen für Maschinenroutinen

Adresse 38786	String search
Adresse 38925	Bubble Sort

```
100 GRAPHICS 0:POKE 710,0:OPEN #1,4,0,"K
:":POKE 752,1                             <HE>
110 POSITION 3,10:? "DISKETTE▲IN▲LAUFWER
K▲1▲UND▲RETURN"                           <OL>
120 GET #1,A:CLOSE #1:OPEN #1,8,0,"D:MEN
U.PIC"                                    <OZ>
130 POSITION 5,12:? "SCHREIBE▲FILE▲==>▲D
:MENU.PIC"                                <HZ>
140 TRAP 300                              <KS>
150 FOR I=1 TO 1214                       <MB>
160 READ A:B=B+A                          <WO>
170 PUT #1,A:IF I/43=INT(I/43) THEN POKE
 40485+P,PEEK(40485+P)+128:P=P+1          <AI>
180 NEXT I                                <GH>
300 POSITION 10,14:IF B<>91431 THEN ? "▲
▲▲DATA▲FEHLER"                            <RS>
310 IF B=91431 THEN ? "FILE▲GESCHRIEBEN"  <HH>
320 ? :? :END                             <MP>
1000 DATA▲216,104,104,133,204,104,133,20
3,104,133,209,104,133,208,104,133,215,10
4,133,214,104,104,133,205,104,104         <BW>
1010 DATA▲133,206,169,0,133,212,169,0,13
3,213,162,0,160,0,177,214,224,0,208,2,13
2,216,209,208,208,43,232,228,206          <OS>
1020 DATA▲240,22,200,196,205,240,50,72,1
52,72,138,168,177,214,133,207,104,168,10
4,165,207,24,144,219,72,165,204           <YU>
1030 DATA▲133,213,165,203,133,212,104,16
2,0,224,0,240,17,224,0,240,6,160,0,177,2
14,162,0,164,216,200,196,205,208          <DP>
1040 DATA▲186,165,208,24,101,205,133,208
,144,2,230,209,165,203,208,6,165,204,240
,7,198,204,198,203,24,144,156,96          <XX>
1050 DATA▲216,104,56,233,3,133,217,104,1
33,204,104,133,203,104,133,215,104,133,2
14,104,133,210,104,133,209,162            <RM>
1060 DATA▲0,104,104,157,0,1,232,228,217,
208,246,56,165,209,233,2,133,209,165,210
,233,0,133,210,48,108,165,209,133         <ZO>
1070 DATA▲211,165,210,133,212,165,204,13
3,206,133,208,165,203,133,205,24,101,214
,133,207,165,208,101,215,133,208          <TT>
1080 DATA▲160,0,185,0,1,190,2,1,134,218,
190,1,1,200,200,200,132,216,168,136,177,
205,209,207,240,12,165,218,208            <TP>
1090 DATA▲4,144,16,176,46,144,44,176,10,
200,202,208,234,164,216,196,217,208,210,
198,211,169,255,197,211,208,6,166         <DN>
1100 DATA▲212,240,11,198,212,165,208,133
,206,165,207,24,144,172,165,213,240,4,13
4,213,208,148,96,134,213,160,0            <MW>
1110 DATA▲177,205,170,177,207,145,205,13
8,145,207,200,196,214,208,241,240,203,0,
0,2,0,0,0,8,0,0,0,0,120,0,0,0,0           <GE>
1120 DATA▲0,62,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,2,0,0,0,16,0,0,0,
0,56,0,0,0,0,0,54,0,0,0,0,112,0           <UV>
1130 DATA▲0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,0,0,0,32,0,0,0,0,56,0,0,0,0,15,6
2,0,0,0,0,56,0,0,0,0,0,0,0,0              <HV>
1140 DATA▲0,0,0,0,0,0,0,0,0,2,0,0,0,64
,15,192,0,0,72,15,192,0,0,13,182,0,0,0,0
,56,0,0,0,31,255,252,0,0,15,255           <YS>
1150 DATA▲240,0,0,0,0,0,0,0,15,128,0,0
,128,8,64,0,0,128,8,64,0,0,15,54,240,0,0
,127,255,248,0,0,24,31,252,0,0            <PH>
1160 DATA▲13,255,176,0,0,30,0,0,0,0,0,7,
0,0,1,0,8,64,0,1,0,8,64,0,0,13,128,192,0
,0,127,255,248,0,0,24,31,240,0            <PY>
1170 DATA▲0,15,255,240,0,0,127,128,0,0,0
,7,226,0,0,18,1,248,127,240,2,1,248,127,
240,0,15,8,192,0,0,12,204,192,0           <QV>
1180 DATA▲0,24,31,240,0,0,13,255,176,0,0
,255,192,0,0,0,4,32,0,0,28,1,8,0,16,4,1,
8,0,16,0,0,8,192,0,0,12,204,192           <QX>
1190 DATA▲0,0,24,31,252,0,0,15,255,240,0
,1,243,224,0,0,0,4,32,0,0,28,1,8,0,16,8,
1,8,0,16,0,0,8,240,0,0,12,204,192         <NK>
1200 DATA▲0,0,31,255,252,0,0,13,255,176,
0,1,225,224,0,0,0,4,63,248,0,30,63,15,25
4,16,16,63,15,254,16,0,0,8,0,0            <LF>
1210 DATA▲0,12,204,192,0,0,31,247,252,0,
0,15,255,240,0,0,3,224,0,0,0,4,0,8,0,0,3
3,0,2,16,0,33,0,2,16,0,0,62,0,0           <HC>
1220 DATA▲0,12,204,192,0,0,31,227,252,0,
```

Listing 1 zu »Happy-Disksorter«. Bitte mit dem Prüfsummer eingeben.

```
0,13,255,176,0,0,7,192,0,0,0,4,0,8,0,0,3
3,0,2,16,0,33,0,2,16,0,0,28,0,0            <RZ>
1230 DATA▴0,12,204,192,0,0,31,193,252,0,
0,15,255,240,0,0,15,128,0,0,0,4,0,8,0,0,
33,255,194,16,0,33,255,194,16,0            <TX>
1240 DATA▴0,8,0,0,0,12,204,192,0,0,31,22
7,252,0,0,15,255,240,0,0,31,0,0,0,0,4,0,
8,0,0,32,0,67,240,0,32,0,67,240            <QU>
1250 DATA▴0,0,0,0,0,0,12,204,192,0,0,31,
247,252,0,0,0,0,0,0,0,62,0,0,0,0,4,0,8,0
,0,32,0,66,0,0,32,0,66,0,0,0,0             <EQ>
1260 DATA▴0,0,0,12,204,192,0,0,31,255,25
2,0,0,127,255,254,0,0,62,7,223,124,0,4,0
,8,0,0,32,0,66,0,0,32,0,66,0,0             <IS>
1270 DATA▴0,0,0,0,0,12,204,192,0,0,31,24
7,252,0,0,96,0,6,0,0,0,6,216,96,0,7,255,
248,0,0,32,0,126,0,0,32,0,126,0            <OS>
1280 DATA▴0,15,188,240,0,0,12,204,192,0,
0,31,227,252,0,0,127,255,254,0,0,62,7,22
3,124,0,0,0,0,0,0,32,0,64,0,0,32           <LG>
1290 DATA▴0,64,0,0,13,182,192,0,0,12,204
,192,0,0,31,227,252,0,0,127,255,254,0,0,
62,6,195,12,0,0,0,0,0,0,32,0,64            <CA>
1300 DATA▴0,0,32,0,64,0,0,15,188,192,0,0
,12,204,192,0,0,31,227,252,0,0,127,255,2
54,0,0,62,6,223,124,0,0,0,0,0,0            <IT>
1310 DATA▴63,255,192,0,0,63,255,192,0,0,
13,182,192,0,0,12,204,192,0,0,31,247,252
,0,0,127,255,254,0,0,0,0,0,0,0             <WM>
1320 DATA▴0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
13,188,240,0,0,15,255,192,0,0,31,255,252
,0,0,12,0,48,0,0,0,0,0,0,32,79             <BD>
1330 DATA▴195,152,15,15,15,15,15,15,15,1
5,15,15,15,15,15,15,15,15,15,15,15,15,66
,64,156,2,2,2,2,2,2,2,2,2,112              <KN>
1340 DATA▴2,112,2,2,2,2,2,2,2,2,2,2,65,1
1,156                                      <IQ>
```

Listing 1 zu »Happy-Disksorter« (Schluß)

```
10 OPEN #2,4,0,"K:"                        <BT>
20 US=ADR("){CTL ,}{CTL M}/{CTL B}Xh"{CT
L P}-Z{CTL C}i{CTL C}{CTL M}R{CTL C}h{CT
L M}U{CTL C}h{CTL M}T{CTL C}h{CTL M}Y{CT
L C}h{CTL M}X{CTL C}▴Vd"{CTL P}){CTL L}{
CTL M}R{CTL C}▴Vd).{CTL M}/{CTL B}{CTL .
}"):P=7                                    <NZ>
40 OPEN #1,4,0,"D:MENU.PIC":I=USR(US,387
86,1214)                                   <YU>
50 POKE 708,10:POKE 709,10:POKE 710,0:PO
KE 82,0:POKE 560,11                        <DC>
200 DIM D$(24000),A$(25)                   <IW>
300 TRAP 300:POKE 559,34:CLOSE #1:POP :E
R=PEEK(195):POKE 195,0:? "{ESC CTL <}{ES
C CTL =}{ESC CTL =}":LD=LEN(D$)            <AD>
310 POKE 752,0:POKE 39977+P,2:POKE 39979
+P,2:? "▴▴▴▴▴Name▴:▴";:INPUT #16;A$:L=LE
N(A$)                                      <BL>
320 IF L=0 THEN 400                        <UN>
325 IF L<25 THEN FOR I=L+1 TO 25:A$(I,I)
="▴":NEXT I                                <GC>
327 D$(LD+1,LD+25)=A$:D$(LD+26,LD+30)="0
00▴▴"                                      <BS>
330 POSITION 1,5:? "Sektor▴#▴:▴";:INPUT
#16;A$:L=LEN(A$):IF L=0 OR L>3 THEN ? "{
ESC CTL 2}";:GOTO 330                      <IB>
350 D$(LD+29-LEN(A$),LD+28)=A$             <DX>
360 POSITION 4,7:? "Index▴:▴";:INPUT #16
;A$:IF LEN(A$)<2 THEN ? "{ESC CTL 2}";:G
OTO 360                                    <JN>
380 D$(LD+29,LD+30)=A$(1,2)                <CJ>
390 GOTO 300                               <MU>
400 POKE 752,1:? "{ESC CTL <}":I=2         <KG>
```

Listing 2 zu »Happy-Disksorter«.
Bitte mit dem Prüfsummer eingeben.
(Fortsetzung auf Seite 74.)

```
410 POSITION I,1:? "{ESC ESC}{ESC CTL -}
":GET #2,A:POSITION I,1:? "▲":I=I+5*(A=4
2 AND I<37)-5*(A=43 AND I>2)               <RA>
430 IF A=155 AND (LD>0 OR I=27 OR I=37)
THEN CNT=LD/30:ON (I-2)/5+1 GOTO 2000,10
00,3000,4000,5000,6000,7000,8000           <UC>
440 GOTO 410                               <NG>
1000 STA=CNT:STO=0:PLMI=-1                 <IP>
1010 D=0:C=D:? "{ESC CTL <}":GOSUB 7600:
? "{ESC CTL =}{ESC CTL =}":POKE 39977+P,
112:POKE 39979+P,112                       <YM>
1020 GOSUB 7400:?                          <WK>
1030 STA=STA+PLMI                          <SX>
1035 IF STA=STO THEN GOSUB 1100:GOTO 300   <RV>
1040 IF PEEK(84)=19 THEN GOSUB 1100:GOTO
 1010                                      <HN>
1050 GOTO 1020                             <OD>
1100 POSITION 3,P+4:? "{ESC ESC}{ESC CTL
 *}":GET #2,A:IF A=32 THEN RETURN          <NS>
1103 IF A=43 OR A=42 THEN STA=STA-PLMI:S
TO=0+(CNT+1)*(STO=0):PLMI=-PLMI:RETURN     <MN>
1105 IF A=27 THEN 300                      <AX>
1107 IF A=155 THEN 1500                    <NC>
1108 POSITION 3,P+4:? "▲"                  <NW>
1110 P=P+1*(A=61 AND P<14)-1*(A=45 AND P
>0):POSITION 3,P+4:? "{ESC ESC}{ESC CTL
*}"                                        <EQ>
1120 FOR I=1 TO 5:POKE 39975+I+P,2+110*(
I=2 OR I=4):NEXT I:GOTO 1100               <RN>
1500 LD=0:POSITION 3,P+4:FOR I=0 TO 2:?
"{ESC CTL +}";:LD=LD+10^I*(PEEK(93)-16):
NEXT I:LD=(LD-1)*30:? "{ESC CTL <}"        <OO>
1505 IF LD<0 THEN 300                      <IS>
1510 POSITION 12,3:? D$(LD+1,LD+25);"{ES
C TAB}{ESC TAB}{ESC TAB}{ESC CTL =}{ESC
CTL *}{ESC CTL *}{ESC CTL *}{ESC CTL *}{
ESC CTL *}";D$(LD+26,LD+28);"{ESC CTL =}
{ESC CTL =}{ESC CTL +}{ESC CTL +}{ESC CT
L +}";D$(LD+29,LD+30):POSITION 0,3:GOTO
310                                        <MR>
2000 TRAP 2020:GOSUB 2500:GOTO 2030        <OO>
2020 GOSUB 2100                            <XP>
2030 STA=STA-P:IF STA<1 THEN STA=1         <YG>
2040 STO=CNT+1:PLMI=1:GOTO 1010            <LF>
2100 F=USR(38786,CNT,ADR(D$),ADR(A$),30,
LEN(A$)):IF F=0 THEN 300                   <BV>
2110 STA=CNT-F+1:RETURN                    <MU>
2500 ? "{ESC CTL <}{ESC CTL =}{ESC CTL =
}{ESC TAB}Gebe▲Term▲oder▲Nummer▲ein{ESC
CTL =}":? "{ESC TAB}===>▲";:INPUT #16;A$
:STA=VAL(A$):IF STA<1 OR STA>CNT THEN 30
0                                          <UK>
2510 RETURN                                <PH>
3000 STA=1:STO=CNT+1:PLMI=1:GOTO 1010      <JD>
4000 ? "{ESC CTL <}"                       <KT>
4010 POSITION 0,6:? "{ESC TAB}1▲=>▲Namen
▲sortieren{ESC CTL =}":? "{ESC TAB}2▲=>▲
Index▲sortieren{ESC CTL =}":GET #2,A:POK
E 559,0                                    <BZ>
4020 IF A=49 THEN I=USR(38925,ADR(D$),30
,CNT,1,28,0,29,2,0)                        <ZE>
4030 IF A=50 THEN I=USR(38925,ADR(D$),30
,CNT,29,2,0,1,28,0)                        <VA>
4040 POKE 559,34:GOTO 300                  <SQ>
5000 TRAP 5020:GOSUB 2500:GOTO 5030        <TQ>
5020 GOSUB 2100                            <XS>
5030 ? "{ESC CTL =}":GOSUB 7400:? :? "{E
SC CTL =}{ESC CTL =}{ESC TAB}{ESC CTL *}
{ESC CTL *}{ESC CTL *}Name▲Loeschen▲(J/N
)":GET #2,A:IF A<>74 THEN 300              <FX>
5100 IF STA=CNT THEN 5200                  <XJ>
5110 D$(STA*30-29,CNT*30-30)=D$(STA*30+1
,CNT*30)                                   <WT>
5200 D$(CNT*30-29)="":CNT=CNT-1:GOTO 500
0                                          <EG>
6000 ? "{ESC CTL <}{ESC CTL =}{ESC CTL =
}{ESC CTL =}{ESC CTL =}{ESC CTL =}{ESC C
TL =}{ESC CTL =}{ESC TAB}1▲=>▲Directory{
ESC CTL =}":? "{ESC TAB}2▲=>▲Speichern{E
SC CTL =}":? "{ESC TAB}3▲=>▲Laden{ESC CT
L =}":? "{ESC TAB}4▲=>▲Dir▲{ESC ESC}{ESC
 CTL *}▲Name":GET #2,A                     <WE>
6040 IF A>48 THEN ON A-48 GOTO 6100,6200
,6200,6500                                 <JX>
6050 GOTO 300                              <OI>
6100 OPEN #1,6,0,"D:*.*"                   <KR>
6110 ? "{ESC CTL <}":? :? ,"Filename▲▲Ex
t▲▲▲▲▲▲Laenge":? ,"{CTL M}{CTL M}{CTL M}
{CTL M}{CTL M}{CTL M}{CTL M}{CTL M}▲▲{CT
L M}{CTL M}▲▲▲▲▲▲{CTL M}{CTL M}{C
TL M}{CTL M}{CTL M}{CTL M}"                <SV>
6120 INPUT #1;A$:IF A$(2,2)<>"▲" THEN ?
:? ,A$:GET #2,A:CLOSE #1:GOTO 6000         <PT>
6130 IF PEEK(53279)=6 THEN 6130            <BV>
6140 POSITION 0,4:? "{ESC SHIFT DEL}":PO
SITION 10,17:? A$(3,10),A$(11,13),A$(15,
17):GOTO 6120                              <AC>
6200 POKE 752,0:? "{ESC CTL =}{ESC CTL =
}Filename▲?▲{CTL Y}(8▲Zeichen){CTL Y}:▲"
;:INPUT #16;A$:POKE 752,1:L=LEN(A$):IF L
=0 OR L>8 THEN 6000                        <OC>
6410 A$(11)=A$(1,L):A$(1,2)="D:":A$(3)=A
$(11,10+L):A$(3+L)=".STR"                  <BA>
6420 IF A=50 THEN OPEN #1,8,0,A$:? #1;LD   <KJ>
6430 IF A=51 THEN OPEN #1,4,0,A$:INPUT #
1;LD:D$="▲":D$(LD-1)=D$:D$(2)=D$           <XA>
6440 I=USR(US,ADR(D$),LD):GOTO 6000        <JB>
6500 POKE 559,0:OPEN #1,6,0,"D:*.*"        <LB>
6510 INPUT #1;A$:IF A$(2,2)<>"▲" THEN PO
KE 559,34:CLOSE #1:GOTO 6000               <DI>
6520 D$(LD+26,LD+28)=A$(15,17):A$=A$(3,1
0):A$(9)="▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲":D$(LD+1,LD+
25)=A$:D$(LD+29)="**"                      <SY>
6530 LD=LD+30:GOTO 6510                    <WM>
7000 ? "{ESC CTL =}{ESC CTL =}{ESC CTL =
}{ESC CTL =}{ESC CTL =}{ESC TAB}1▲==>▲Dr
ucke▲2▲Spalten{ESC CTL =}":? "{ESC TAB}2
▲==>▲Drucke▲3▲Spalten"                     <PH>
7020 GET #2,A:IF A<49 OR A>50 THEN 300     <VN>
7030 D=1:C=0:OPEN #D,4,0,"P:":IF A=50 TH
EN 7100                                    <SJ>
7040 ? #D;"{ESC ESC}!{CTL ,}":GOSUB 7600
:GOSUB 7600:? #D:? #D                      <XV>
7045 STO=INT((CNT-0.1)/2)+1                <NF>
7050 FOR I=1 TO CNT                        <TK>
7060 STA=I:GOSUB 7400:STA=STO+I:GOSUB 74
00:? #D:NEXT I:GOTO 300                    <GT>
7100 ? #D;"{ESC ESC}!{CTL D}":GOSUB 7600
:GOSUB 7600:GOSUB 7600:? #D:? #D           <AQ>
7110 STO=INT(CNT/3):R=CNT/3-STO            <TN>
7120 FOR I=1 TO CNT:S2=I+STO+1*(R>0)       <JL>
7130 STA=I:GOSUB 7400:STA=S2:GOSUB 7400:
STA=S2+STO+1*(R>0.5):GOSUB 7400:? #D:NEX
T I                                        <WU>
7400 GOSUB 7500:? #D;A$;"▲";D$(LD-29,LD-
5);"▲▲";D$(LD-4,LD-2);"▲";D$(LD-1,LD);"▲
▲";:RETURN                                 <FA>
7500 C=C+1:IF C>CNT THEN ? #D:CLOSE #1:G
OTO 300                                    <TZ>
7510 A$="000":A$(4-LEN(STR$(STA)))=STR$(
STA):LD=STA*30:RETURN                      <XI>
7600 ? #D;"Nr.▲▲▲▲▲>>>▲Programm▲<<<▲▲▲▲▲
▲▲Sek▲In▲▲";:RETURN                        <AV>
8000 TRAP 9200:POKE 39982,7:POSITION 0,9
:? "▲▲▲";:GOSUB 9000+ER                    <XJ>
8002 ? :? "{ESC TAB}{ESC TAB}{ESC TAB}{E
SC TAB}Druecke▲eine▲Taste":GET #2,A:POKE
 39982,2:POKE 195,0:GOTO 300               <KM>
9000 ? "▲▲KEIN▲FEHLER":RETURN              <SA>
9137 ? "▲▲FILE▲FEHLER":RETURN              <PX>
9138 ? "▲GERAET▲FEHLT":RETURN              <MO>
9144 ? "FEHLER▲IM▲GERAET":RETURN           <BR>
9162 ? "▲DISKETTE▲VOLL":RETURN             <NI>
9163 ? "▲SYSTEM▲FEHLER":RETURN             <HQ>
9165 ? "FILENAME▲FEHLER":RETURN            <FX>
9167 ? "FILE▲GESCHUEZT":RETURN             <EI>
9169 ? "DIRECTORY▲VOLL":RETURN             <NE>
9170 GOTO 9165                             <XH>
9173 ? "FORMAT▲SCHLECHT":RETURN            <YT>
9200 ? "▲▲FEHLER▲";ER:GOTO 8002            <GI>
```

Listing 2 zu »Happy-Disksorter« (Schluß)

Submission – ein Weg voller Gefahren

Böse Zungen behaupten, es gibt keine Helden mehr. Falsch! Auf gar mancher Diskette eines Atari 800XL/130XE kämpfen sie für das hohe Ideal der Freiheit.

Nur die Besten werden überleben. Unter diesem Motto gilt es sich zu beweisen, denn in den tiefen Gängen des Labyrinths im Spiel »Submission« lauert der Tod mit seinen tausend Gesichtern.

Ziel der Mission ist die Befreiung einer Frau. Der Weg zu ihr führt durch verschiedene Ebenen, die durch Leitern miteinander verbunden sind. Das Klettern ist dabei noch das einfachste, aber es wäre kein Action-Spiel, gäbe es nicht viele Gefahren, die einem das Leben schwermachen. Da plagt man sich mit Krokodilen, Spinnen und Minen, auch an Selbstschußanlagen fehlt es nicht. Unterwegs kann man Goldschätze aufsammeln, für die es natürlich Punkte gibt. Pro gelungener Befreiungsaktion gibt es dann einen Bonus. Die Abwechslung kommt nicht zu kurz, denn je nach Belieben stehen sieben verschiedene Schwierigkeitsgrade zur Verfügung. Ein weiterer Pluspunkt ist die leichte Handhabung des Spiels. Die einzelnen Spielvarianten ändert man über die SELECT-Taste. Durch Drücken der START-Taste ist man gleich mitten im Geschehen und nur noch auf die meisterliche Handhabung des Joysticks angewiesen.

»Submission« ist zu 100 Prozent in Maschinensprache geschrieben. Daraus ergibt sich auch die Schnelligkeit des Spielablaufs. Bevor Sie jedoch das Listing zu »Submission« eingeben, müssen Sie unbedingt den AMPEL aus diesem Sonderheft abtippen. Das erleichtert Ihnen die Eingabe von Atari-Maschinensprach-Programmen. Automatisch werden sämtliche Tippfehler erkannt und auf dem Bildschirm angezeigt.

(Peter Raab/wb)

Peter Blümer, der Autor des Spiels »Submission«, ist unseren Lesern schon bekannt. In Happy-Computer, Ausgabe 1/85, veröffentlichten wir bereits ein anderes, allerdings in Basic programmiertes, Spiel »Die Schatzhöhle«. Mittlerweile ist für ihn das Wort Maschinensprache kein Fremdwort mehr, wie das Spiel »Submission« beweist. Die Umsetzung von der Idee zum tatsächlichen Spiel ist vorzüglich gelungen. Obwohl das zugehörige Listing entsprechend lang ist, wird sich der Aufwand, es in den Atari einzugeben, wirklich lohnen.

PROGRAMM-STECKBRIEF	
Programmname	Submission
Programmtyp	Spiel
Programmiersprache	Assembler
Programmlänge	11 688 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk oder Kassetten-Recorder
Eingabehilfe	AMPEL
Bemerkung	Labyrinth-Spiel
Leserservice	Diskette (SUBMISS.COM)

```
0000:FF FF 00 06 BA 06 A9 00<4F>
0008:8D 44 02 85 41 8D C6 02<22>
0010:8D C5 02 8D C8 02 AD 30<E8>
0018:02 85 CE AD 31 02 85 CF<72>
0020:A0 0B A9 46 91 CE C8 A9<A2>
0028:57 91 CE C8 A9 06 91 CE<CA>
0030:C8 A9 06 91 CE C8 A9 07<36>
0038:91 CE C8 A9 06 91 CE C8<8F>
0040:A9 06 91 CE A9 36 8D C4<00>
0048:02 A9 01 85 09 A0 50 A2<39>
0050:06 A9 06 4C 5C E4 D8 EE<43>
0058:C7 02 4C 5F E4 00 00 00<BA>
0060:00 00 00 2C 2F 21 24 29<92>
0068:2E 27 00 00 00 00 00 00<09>
0070:00 00 00 00 00 00 00 00<70>
0078:00 00 00 00 00 00 00 00<78>
0080:00 00 00 00 00 00 00 00<80>
0088:00 00 F3 F5 E2 ED E9 F3<BD>
0090:F3 E9 EF EE 00 00 00 00<01>
0098:00 00 00 00 00 00 00 00<98>
00A0:00 00 00 00 00 00 00 00<A0>
00A8:00 00 00 00 00 00 00 22<CA>
00B0:39 00 30 25 34 25 32 00<3E>
00B8:22 2C 35 25 2D 25 32 00<2E>
00C0:00 E2 02 E3 02 00 06 0C<0E>
00C8:00 0D 00 00 22 00 22 FB<5C>
00D0:22 A0 5F A2 E4 A9 06 20<CD>
00D8:5C E4 4C 72 49 00 00 00<8A>
00E0:00 00 00 00 00 00 00 00<E0>
00E8:00 00 00 00 00 00 00 00<E8>
00F0:00 00 00 00 00 00 00 00<F0>
00F8:00 00 00 00 00 00 00 00<F8>
0100:00 00 00 00 00 00 00 00<02>
0108:00 00 00 00 00 00 00 00<0A>
0110:00 00 00 00 00 00 00 00<12>
0118:00 00 00 00 00 00 00 00<1A>
0120:00 00 00 00 00 00 00 00<22>
0128:00 00 00 00 00 02 02 02<38>
0130:02 02 02 02 02 02 02 02<32>
0138:02 02 02 02 02 02 02 02<3A>
0140:01 03 03 03 03 03 03 03<41>
0148:03 03 03 03 03 03 03 03<4A>
0150:03 03 03 03 03 00 00 00<3D>
0158:00 00 00 00 00 00 00 00<5A>
0160:00 00 00 00 00 00 00 00<62>
0168:00 00 00 00 00 00 00 00<6A>
0170:00 00 00 00 00 00 00 00<72>
0178:00 00 00 00 00 00 00 00<7A>
0180:00 00 00 00 00 00 00 00<82>
0188:00 00 00 00 00 00 00 00<8A>
0190:00 00 00 00 00 00 00 00<92>
0198:00 00 00 00 00 00 00 00<9A>
01A0:00 00 00 00 00 00 00 00<A2>
01A8:00 00 00 00 00 44 44 00<42>
01B0:47 48 02 02 02 02 02 02<E6>
01B8:01 00 00 00 00 00 00 00<3E>
01C0:00 45 45 00 C9 4A 03 03<31>
01C8:03 03 03 03 03 FC 22 F7<E0>
01D0:23 00 00 00 00 00 00 00<64>
01D8:00 45 45 00 4B 4C 02 02<62>
01E0:02 02 02 02 01 00 00 00<CC>
01E8:00 00 00 00 00 46 46 00<8C>
01F0:CE CD 03 03 03 03 03 03<CA>
01F8:03 02 02 02 01 00 00 00<65>
0200:00 00 00 00 00 00 00 00<04>
0208:00 00 00 00 00 03 03 03<21>
0210:03 00 00 00 00 00 00 00<95>
0218:00 00 00 00 00 00 00 00<1C>
0220:00 02 02 02 01 00 00 00<0D>
0228:00 00 00 00 00 00 00 00<2C>
0230:00 00 00 00 00 03 03 03<49>
0238:03 00 00 00 00 00 00 00<BD>
0240:00 00 00 00 00 00 00 00<44>
0248:00 02 02 02 01 00 00 00<35>
0250:00 00 00 00 00 00 00 00<54>
0258:00 00 00 00 00 03 03 03<71>
0260:03 00 00 00 00 00 00 00<E5>
0268:00 00 00 00 00 00 00 00<6C>
0270:00 02 02 02 02 02 02 02<73>
0278:02 02 02 02 02 02 02 02<7C>
0280:02 02 02 02 01 03 03 03<83>
0288:03 03 03 03 03 03 03 03<8C>
0290:03 03 03 03 03 03 03 03<94>
0298:03 00 00 00 00 9F 00 A0<38>
02A0:00 9F 00 00 00 00 00 00<4C>
02A8:00 02 02 02 01 00 00 00<95>
02B0:00 9F 00 A0 00 9F 00 00<D0>
02B8:00 00 00 00 00 03 03 03<D1>
02C0:03 00 00 00 00 00 00 00<46>
02C8:00 00 00 00 00 F8 23 F3<E5>
02D0:24 00 00 00 00 02 02 02<F4>
02D8:01 00 00 00 00 00 00 00<5D>
02E0:00 00 00 00 00 00 00 00<E4>
02E8:00 03 03 03 03 00 00 00<56>
02F0:00 00 00 00 00 00 00 00<F4>
02F8:00 00 00 00 00 02 02 02<0B>
0300:01 00 00 00 00 00 00 00<86>
0308:00 00 00 00 00 00 00 00<0E>
0310:00 00 DE 03 03 00 00 00<1A>
0318:00 00 00 00 00 00 00 00<1E>
0320:00 00 00 00 00 00 00 00<26>
0328:00 28 25 2C 30 0E 0E 0E<7B>
0330:0E 00 00 00 00 00 00 00<3D>
0338:00 00 00 00 00 00 00 00<3E>
0340:00 00 00 00 00 00 00 00<46>
0348:00 00 00 00 00 00 00 00<4E>
0350:00 00 00 00 00 00 00 00<56>
0358:02 02 02 02 02 02 02 02<3E>
0360:02 02 02 02 02 02 02 02<66>
0368:02 02 02 01 03 03 03 03<6D>
0370:03 03 03 03 03 03 03 03<76>
0378:03 03 03 03 03 03 03 03<3E>
0380:00 00 00 00 00 9F 00 00<01>
0388:00 00 00 00 00 00 00 A0<2E>
```

Listing zu »Submission«. Bitte mit AMPEL eingeben.

```
0390:00 00 00 00 00 00 00 00<96>
0398:00 9F 00 00 00 00 00 00<46>
03A0:00 00 00 A0 00 00 00 00<A0>
03A8:02 02 02 02 01 00 00 00<98>
03B0:44 44 00 47 48 02 02 02<A6>
03B8:02 02 01 03 03 03 03 03<7D>
03C0:03 00 00 00 45 45 00 C9<50>
03C8:4A 03 03 03 03 F4 24 EF<64>
03D0:25 03 03 03 02 02 02 02<DB>
03D8:01 00 00 00 45 45 00 4B<E1>
03E0:4C 02 02 02 02 02 02 01<8A>
03E8:03 03 03 03 03 00 00 00<D9>
03F0:46 46 00 CE CD 03 03 03<8B>
03F8:03 03 03 03 02 02 02 02<6F>
0400:02 02 02 02 02 02 02 02<08>
0408:02 02 02 02 02 02 02 01<0F>
0410:03 03 03 03 03 03 03 03<18>
0418:03 03 03 03 03 03 03 03<20>
0420:03 03 03 03 02 02 02 02<19>
0428:02 02 02 02 02 02 02 02<30>
0430:02 02 02 02 02 02 02 01<37>
0438:03 03 03 03 03 03 03 03<40>
0440:03 03 03 03 03 03 03 03<48>
0448:03 03 03 03 02 02 02 02<41>
0450:01 00 00 00 A0 00 00 9F<7C>
0458:00 00 00 02 02 02 02 01<9D>
0460:03 03 03 03 03 00 E1 00<15>
0468:A0 00 00 9F 00 E1 00 03<C0>
0470:03 03 03 03 02 02 02 02<69>
0478:01 00 00 00 A0 00 00 9F<A5>
0480:00 00 00 00 02 02 02 01<C5>
0488:03 03 03 03 03 00 00 00<7B>
0490:A0 00 00 9F 00 00 00 03<D5>
0498:03 03 03 03 00 00 00 9F<12>
04A0:00 A0 00 9F 00 A0 00 9F<A8>
04A8:00 A0 00 9F 00 00 00 00<82>
04B0:00 00 00 9F 00 A0 00 9F<C0>
04B8:00 A0 00 9F 00 A0 00 9F<F0>
04C0:00 00 00 00 00 00 00 9F<67>
04C8:9F 00 00 A0 A0 F0 25 EB<24>
04D0:26 00 00 A0 A0 00 00 9F<81>
04D8:9F 00 00 00 00 00 00 9F<CF>
04E0:9F 00 00 A0 A0 00 00 A0<D7>
04E8:A0 00 00 9F 9F 00 00 00<B7>
04F0:00 00 00 00 00 00 00 00<F8>
04F8:00 00 00 00 00 00 00 00<00>
0500:02 02 02 01 00 00 00 00<DB>
0508:00 00 00 00 00 00 00 00<12>
0510:00 00 00 00 03 03 03 03<47>
0518:02 02 02 02 02 02 02 02<22>
0520:02 02 02 02 02 02 02 02<2A>
0528:02 02 02 01 03 03 03 03<31>
0530:03 03 03 03 03 03 03 03<3A>
0538:03 03 03 03 03 03 03 03<42>
0540:02 02 02 02 02 02 02 02<4A>
0548:02 02 02 02 02 02 02 02<52>
0550:02 02 02 01 03 03 03 03<59>
0558:03 03 03 03 03 03 03 03<62>
0560:03 03 03 03 03 03 03 03<6A>
0568:00 00 00 00 00 00 00 00<72>
0570:00 00 00 00 00 00 00 00<7A>
0578:00 00 00 00 00 00 00 00<82>
0580:00 00 00 00 00 00 00 00<8A>
0588:00 00 00 00 00 00 00 00<92>
0590:00 00 00 00 00 00 00 00<9A>
0598:00 00 00 00 00 00 00 00<A2>
05A0:00 00 00 00 00 00 00 00<AA>
05A8:00 00 00 00 00 00 00 00<B2>
05B0:00 00 00 00 00 00 00 00<BA>
05B8:00 00 00 00 00 00 00 00<C2>
05C0:00 00 00 02 01 00 00 00<1B>
05C8:00 00 00 00 00 EC 26 E7<B5>
05D0:27 00 00 00 00 00 00 00<6E>
05D8:02 01 03 03 00 00 00 00<B4>
05E0:00 00 00 00 00 00 00 00<EA>
05E8:00 00 00 00 03 03 02 01<1C>
05F0:00 00 00 00 00 00 00 00<FA>
05F8:00 00 00 00 00 00 00 00<02>
0600:02 01 03 03 00 00 00 00<DD>
0608:00 00 00 00 00 00 00 00<14>
0610:00 00 00 00 03 03 00 00<40>
0618:00 00 00 00 00 00 00 00<24>
0620:00 00 00 00 00 00 00 00<2C>
0628:00 00 00 00 00 00 00 00<34>
0630:00 00 00 00 00 00 00 00<3C>
0638:00 00 00 00 00 00 00 00<44>
0640:00 00 00 00 00 00 00 00<4C>
0648:00 00 00 00 00 00 00 00<54>
0650:00 00 00 00 00 00 00 00<5C>
0658:00 00 00 00 00 00 00 00<64>
0660:00 00 00 00 00 00 00 00<6C>
0668:00 00 00 00 00 00 00 00<74>
0670:00 00 00 00 00 00 00 00<7C>
0678:00 00 02 02 02 02 01 00<FE>
0680:00 00 47 48 00 44 44 00<93>
0688:00 00 00 00 00 00 03 03<9D>
0690:03 03 03 00 00 00 C9 4A<1A>
0698:00 45 45 00 00 00 00 00<9E>
06A0:00 00 02 02 02 02 01 00<27>
06A8:00 00 4B 4C 00 45 45 00<70>
06B0:00 00 00 00 00 00 03 03<C5>
06B8:03 03 03 00 00 00 CE CD<D0>
06C0:00 46 46 00 00 00 00 00<07>
06C8:00 00 00 00 00 E8 27 34<F6>
06D0:28 00 00 00 00 00 00 00<F0>
06D8:00 00 00 00 00 00 00 00<E4>
06E0:00 00 00 00 00 00 00 00<EC>
06E8:00 00 00 00 00 00 00 00<F4>
06F0:00 00 00 00 00 00 02 02<03>
06F8:02 01 00 00 00 00 00 00<45>
0700:00 A0 00 00 9F 00 00 00<2B>
0708:00 00 03 03 03 03 00 00<CA>
0710:00 00 00 00 00 A0 00 00<9C>
0718:9F 00 00 00 00 00 35 28<86>
0720:2E 29 E6 1B 3D 50 1B BA<15>
0728:E9 1E 53 88 BB E6 1B E6<9E>
0730:F0 23 88 23 58 BC 58 1E<94>
0738:87 BD 23 88 87 23 BC 58<66>
0740:F0 EF 59 24 BD 87 53 88<A0>
0748:BD 88 23 88 BC E9 D7 0A<AC>
0750:50 EF F0 23 56 BC 89 F1<A9>
0758:56 23 BC 89 F1 26 5B 50<10>
0760:BD 23 58 BC 88 87 90 59<20>
0768:24 BD 5B BD 90 59 24 87<4B>
0770:23 BC 58 88 BD 53 5B 26<30>
0778:23 F1 89 BC 23 2D C5 56<CE>
0780:F1 89 BC 23 F1 89 F8 26<D2>
0788:F1 89 BC 95 23 95 A6 2D<15>
0790:C4 F9 F1 89 62 56 A6 95<EC>
0798:2D C4 C4 F9 23 F1 F8 40<43>
07A0:41 49 41 41 41 41 42 42<AF>
07A8:42 42 40 41 40 42 43 44<58>
07B0:43 43 43 43 42 43 44 43<B4>
07B8:44 43 43 43 43 42 43 44<1B>
07C0:44 44 43 42 42 44 44 43<34>
07C8:44 43 41 48 49 41 43 44<46>
07D0:45 45 45 45 45 45 45 45<56>
07D8:45 45 46 46 41 44 43 43<44>
07E0:43 44 43 44 44 44 44 46<E7>
07E8:44 46 44 44 43 43 43 43<DF>
07F0:44 44 42 46 46 45 45 45<6B>
07F8:45 45 47 46 45 45 45 45<2D>
0800:45 45 45 46 46 45 45 45<D3>
0808:47 45 47 48 47 47 47 45<63>
0810:45 47 45 48 47 47 47 47<DA>
0818:47 45 45 46 2F 29 2A 2A<A2>
0820:70 70 70 46 5C 22 06 C6<EC>
0828:56 29 07 06 06 06 06 02<45>
0830:02 02 02 02 02 C6 34 22 06<6B>
0838:06 06 46 CF 24 06 06 06<CD>
0840:46 5C 22 06 41 2F 29 00<FF>
0848:00 00 00 00 00 00 00 00<58>
0850:00 00 00 00 00 00 00 00<60>
0858:00 00 00 00 00 00 00 00<68>
0860:F3 F5 E2 ED E9 F3 F3 E9<14>
0868:EF EE 00 00 00 00 00 00<6B>
0870:00 00 00 00 00 00 00 00<80>
0878:00 00 00 00 00 00 00 00<88>
0880:00 00 00 00 00 00 22 39 00<8B>
0888:30 25 34 25 32 00 22 2C<B4>
0890:35 25 2D 25 32 00 00 00<CD>
0898:00 00 23 2F 30 39 32 29<EB>
08A0:27 28 34 00 11 19 18 15<06>
08A8:00 00 00 00 00 00 00 00<B8>
08B0:00 00 00 00 00 00 00 00<C0>
08B8:00 00 00 00 00 00 00 2C<F4>
08C0:21 33 34 0D 33 23 2F 32<3A>
08C8:25 1A 00 00 00 00 00 00<F1>
08D0:00 00 00 00 00 28 29 27<FA>
08D8:28 0D 33 23 2F 32 25 1A<3F>
08E0:10 10 10 10 10 10 10 00<D8>
08E8:00 00 00 00 00 00 00 00<F8>
08F0:00 00 00 00 00 00 00 00<00>
08F8:00 00 00 00 00 00 00 00<08>
0900:00 00 00 00 00 00 00 00<12>
0908:00 00 00 00 00 00 00 00<1A>
0910:00 00 30 32 25 33 33 00<A7>
0918:33 25 2C 25 2B 2A 26 2B<5E>
0920:23 34 00 34 2F 00 23 28<FB>
0928:21 2E 27 25 00 33 34 21<E3>
0930:32 34 29 2E 27 00 33 35<34>
0938:22 00 00 00 00 00 00 00<5B>
0940:00 00 00 00 00 00 00 00<52>
0948:00 00 00 00 00 00 11 00<7C>
0950:00 00 00 00 00 00 00 00<62>
0958:00 00 00 00 00 00 00 00<6A>
0960:00 00 00 00 00 00 00 00<72>
0968:00 30 32 25 33 33 00 33<70>
0970:34 21 32 34 00 34 2F 00<9D>
0978:30 2C 21 39 00 2E 25 37<9F>
0980:00 27 21 2D 25 00 00 00<5C>
0988:00 00 00 00 00 00 00 00<9A>
0990:00 00 00 00 00 00 00 00<A2>
0998:00 00 00 00 00 00 00 00<AA>
09A0:00 00 00 00 00 00 00 00<B2>
09A8:00 00 00 00 00 00 00 00<BA>
09B0:00 00 00 00 00 00 00 00<C2>
09B8:A9 00 8D 2F 02 8D C5 02<71>
09C0:8D C8 02 8D C6 02 A9 40<DD>
09C8:8D 0E D4 A2 00 BD 2F 29<D3>
09D0:9D 86 06 E8 E0 4B D0 F5<92>
09D8:A9 80 8D F4 02 A9 A6 8D<88>
09E0:C4 02 A9 86 8D 30 02 A9<9C>
09E8:06 8D 31 02 A2 00 BD 5D<4C>
09F0:48 9D D9 29 E8 E0 08 D0<99>
09F8:F5 A9 6E 8D 00 02 A9 2B<09>
0A00:8D 01 02 A9 C0 8D 0E D4<06>
0A08:A9 22 8D 2F 02 AD 1F D0<CE>
0A10:C9 05 D0 1B AD 78 30 C9<E2>
0A18:00 D0 14 A9 27 2B B5 2B<B7>
0A20:01 8D 78 30 EE 59 2A AD<9F>
0A28:59 2A C9 18 D0 05 A9 11<04>
0A30:8D 59 2A AD 1F D0 C9 07<C1>
0A38:D0 05 A9 00 8D 78 30 EE<23>
0A40:79 30 AD 79 30 C9 FF D0<3C>
0A48:08 A9 00 8D 79 30 20 E6<04>
0A50:35 AD 78 30 C9 00 D0 B9<DB>
0A58:AD 1F D0 C9 06 D0 B2 A9<8E>
0A60:01 8D 78 30 4C 7E 49 48<40>
0A68:86 CC AD 0B D4 C9 3C B0<6A>
0A70:1F A9 78 8D 09 D4 A2 00<11>
0A78:A5 CD 8D 0A D4 8D 19 D0<97>
0A80:E6 CD E8 E0 0F D0 F1 A9<38>
0A88:0E 8D 17 D0 A6 CC 68 40<13>
0A90:A9 A4 8D 18 D0 A9 80 8D<48>
0A98:0A D4 8D 09 D4 A9 34 8D<27>
0AA0:16 D0 A9 16 8D 17 D0 A9<3B>
0AA8:1E 8D 19 D0 4C 93 2B B6<10>
0AB0:2B B1 2C AF AF AC AA AB<1A>
0AB8:A5 A2 A0 0A 00 00 00 01<3D>
0AC0:00 00 00 00 0A 00 00 01<76>
0AC8:00 01 00 0A 00 00 00 01<BE>
0AD0:00 00 00 0A 01 0A 01 01<B8>
0AD8:01 01 01 0C 0C 0A 08 06<2D>
0AE0:04 00 F3 F3 60 60 79 79<70>
0AE8:A2 A2 90 90 60 60 79 79<4F>
0AF0:90 90 B6 B6 5B 5B 79 79<36>
0AF8:90 90 A2 A2 51 51 6C 6C<D3>
0B00:80 80 51 00 51 00 51 00<AD>
0B08:51 00 60 00 60 00 60 00<94>
0B10:60 00 5B 00 5B 00 60 00<3D>
0B18:5B 00 6C 6C 6C 6C 6C 6C<55>
0B20:6C 00 79 79 79 00 79 79<5A>
0B28:79 00 90 90 90 00 90 90<13>
0B30:90 00 B6 B6 B6 00 B6 B6<F7>
0B38:B6 00 A2 A2 A2 A2 A2 A2<14>
0B40:A2 00 A2 00 A2 00 A2 00<CD>
0B48:A2 00 90 00 90 00 90 00<C4>
0B50:90 00 79 00 79 00 79 00<12>
0B58:79 00 6C 6C 6C 6C 6C 6C<4C>
0B60:6C 00 60 00 60 00 60 00<FB>
0B68:60 00 90 00 90 00 90 00<C5>
0B70:90 00 5B 5B 5B 00 5B 5B<D4>
0B78:5B 00 80 80 80 80 80 80<A3>
0B80:80 00 EA A0 AB A2 2C A9<F9>
0B88:07 20 5C E4 A9 C0 8D 0E<61>
0B90:D4 A9 00 8D 08 D2 8D 7D<E5>
0B98:30 8D 7C 30 8D 7A 30 8D<F8>
0BA0:7B 30 A9 03 8D 0F D2 60<4B>
0BA8:D8 86 CE AE 7A 30 BD B2<60>
0BB0:2C 63 2D E5 2B 8D 00 D2<07>
0BB8:AE 7C 30 BD E5 2B C9 00<BD>
0BC0:F0 06 BD B6 2B 8D 01 D2<A6>
0BC8:AD 7C 30 C9 07 D0 29 A9<19>
0BD0:00 8D 7C 30 8D 03 D2 EE<9E>
0BD8:7A 30 EE 7B 30 AD 7A 30<84>
0BE0:C9 20 D0 14 A9 00 8D 7A<99>
```

Listing zu »Submission« (Fortsetzung)

```
0BE8:30 AD 7D 30 C9 00 D0 08<62>
0BF0:A9 00 8D 7B 30 EE 7D 30<76>
0BF8:EE 7C 30 AD 7D 30 C9 00<2B>
0C00:F0 23 AE 7B 30 BD 05 2C<E0>
0C08:8D 06 D2 18 69 01 8D 02<8E>
0C10:D2 A9 A4 8D 03 D2 8D 07<3A>
0C18:D2 AD 7B 30 C9 80 D0 05<A3>
0C20:A9 00 8D 7B 30 AE 7A 30<43>
0C28:BD BE 2B 8D 04 D2 C9 00<49>
0C30:F0 09 AE 7C 30 BD DE 2B<6B>
0C38:8D 05 D2 A9 02 8D 08 D2<D5>
0C40:A6 CE 4C 62 E4 A0 62 A2<34>
0C48:E4 A9 07 20 5C E4 A9 C0<1D>
0C50:8D 0E D4 A2 00 A9 00 9D<06>
0C58:00 D2 E8 E0 09 D0 F8 A9<0C>
0C60:03 8D 0F D2 60 64 2D 5F<66>
0C68:2E 00 00 00 3F 7F F8 F0<67>
0C70:E0 C0 C0 C0 60 00 1C 3C<62>
0C78:26 0E 1E 1E 0E 26 3E 3C<8C>
0C80:1C 00 00 00 00 00 60 70<D7>
0C88:3E 1E 00 00 00 00 00 00<07>
0C90:00 00 06 0A 1F 2E 30 3E<58>
0C98:18 18 00 00 18 30 60 60<20>
0CA0:30 18 00 00 00 18 14 12<71>
0CA8:12 14 18 10 00 00 00 00<D2>
0CB0:00 00 00 00 3E 7F F8 F0<94>
0CB8:E0 C0 C0 C0 60 00 1C 3C<5A>
0CC0:2E 26 32 78 7C 7E 7E 3C<40>
0CC8:18 00 00 00 00 80 81 C7<B8>
0CD0:E7 66 00 00 00 00 00 00<F5>
0CD8:00 00 06 0A 1F 3E 30 3E<DE>
0CE0:18 18 00 00 10 18 0C 06<89>
0CE8:03 01 00 00 28 38 28 2C<5E>
0CF0:24 64 44 00 00 00 00 00<BB>
0CF8:00 00 00 00 FC FE 1F 0F<3B>
0D00:07 03 03 03 06 00 38 3C<CB>
0D08:64 70 78 78 70 64 7C 3C<34>
0D10:38 00 00 00 00 00 06 0E<60>
0D18:7C 78 00 00 00 00 00 00<4E>
0D20:00 00 60 50 F8 74 0C 7C<4C>
0D28:18 18 00 00 18 0C 06 06<57>
0D30:0C 18 00 00 00 18 28 48<4F>
0D38:48 28 18 08 00 00 00 00<C3>
0D40:00 00 00 00 FC FE 1F 0F<7F>
0D48:07 03 03 03 06 00 1C 3C<DB>
0D50:34 24 0C 1E 3E 3E 3E 3E<05>
0D58:38 00 00 00 00 01 81 E3<F8>
0D60:E7 66 00 00 00 60 2E 5B<7C>
0D68:2F 00 00 00 00 00 60 50<2A>
0D70:F8 74 0C 7C 18 18 00 00<0E>
0D78:08 18 30 60 C0 80 00 00<8C>
0D80:1C 1C 1C 36 22 22 22 02<72>
0D88:02 00 00 00 00 00 00 00<A3>
0D90:FC FE 3F 1F 0F 07 03 03<9F>
0D98:06 00 3C 34 26 0E 1E 3E<52>
0DA0:1F 0F 07 03 02 80 F8 78<95>
0DA8:00 00 00 00 00 00 00 00<C2>
0DB0:00 00 00 00 00 00 00 00<CA>
0DB8:00 00 00 00 40 60 50 F8<EE>
0DC0:7C 0C 7C 18 00 08 18 30<0C>
0DC8:E0 C0 E0 B0 98 DC 6C 30<16>
0DD0:00 00 00 00 00 00 00 00<EA>
0DD8:00 00 00 00 00 00 3F 7F<F0>
0DE0:FC F8 F0 E0 C0 C0 60 00<B2>
0DE8:3C 2C 64 70 78 7C F8 F0<48>
0DF0:E0 C0 40 01 1F 1E 00 00<F3>
0DF8:00 00 00 00 00 00 00 00<12>
0E00:00 00 00 00 00 00 00 00<1C>
0E08:00 00 00 02 06 0A 1F 3E<18>
0E10:30 3E 18 00 10 18 0C 07<D6>
0E18:03 07 0D 19 3B 36 0C 00<75>
0E20:00 00 00 00 00 00 00 00<3C>
0E28:00 00 00 00 00 00 00 00<44>
0E30:00 00 7E FF FF 7E 7E FF<F7>
0E38:7E 24 00 7E 3C 3C 7E 7E<41>
0E40:3C 3C 3C 3C 3C 00 00 00<76>
0E48:40 40 60 60 08 08 0C 0C<8A>
0E50:00 00 00 00 00 00 00 00<6C>
0E58:00 81 81 00 18 19 19 01<FC>
0E60:43 C2 80 80 80 5C 2F BD<9E>
0E68:2F 80 80 00 00 3C 3C 24<B9>
0E70:24 24 04 04 04 04 04 00<A0>
0E78:00 00 00 00 00 00 7E FF<90>
0E80:FF 7E 7E FF 7E 24 00 7E<5D>
0E88:3C 3C 7E 7E 3C 3C 3C 3C<EC>
0E90:3C 00 00 00 08 08 0C 0C<4F>
0E98:40 40 60 60 00 00 00 00<F6>
0EA0:00 00 00 00 00 81 81 00<C5>
0EA8:98 98 98 80 C2 43 01 01<C3>
0EB0:01 01 01 00 00 3C 3C 24<3B>
0EB8:24 24 20 20 20 20 20 00<B7>
0EC0:00 00 00 00 00 00 00 00<DC>
0EC8:00 00 00 BE 2F B9 30 E2<5F>
0ED0:E2 E6 E3 E4 E5 E6 E2 E1<55>
0ED8:30 30 30 60 30 90 60 30<45>
0EE0:96 55 96 55 7C 7C 96 34<42>
0EE8:D8 58 F8 96 0A 34 94 54<4A>
0EF0:76 D8 08 A8 16 80 32 D2<E0>
0EF8:00 40 00 80 00 00 00 00<2C>
0F00:00 00 50 50 50 00 50 00<2E>
0F08:00 00 00 00 50 00 00 00<A0>
0F10:00 50 00 00 00 5A 00 00<6B>
0F18:00 00 00 00 00 50 00 00<77>
0F20:50 00 00 00 50 00 50 00<F8>
0F28:00 00 00 00 00 00 00 50<96>
0F30:00 00 00 00 5A 00 00 00<19>
0F38:00 00 00 00 00 00 50 00<F6>
0F40:00 00 5A 00 00 00 00 00<89>
0F48:5A 00 00 00 50 50 00 46<1C>
0F50:00 00 00 00 00 50 00 50<FF>
0F58:00 00 00 00 50 00 00 00<F0>
0F60:50 00 00 00 50 00 00 00<A0>
0F68:00 50 00 00 00 00 00 00<9A>
0F70:00 00 50 00 00 00 00 00<98>
0F78:50 00 00 E1 00 A0 00 00<4F>
0F80:00 5A 30 00 00 00 00 00<3B>
0F88:00 00 00 00 00 00 00 00<A6>
0F90:00 00 00 00 00 00 00 00<AE>
0F98:00 00 00 00 01 00 1E 1C 01<78>
0FA0:1A 00 1A 00 00 00 02 04<17>
0FA8:06 00 08 00 FF 7C 6E 6E<FB>
0FB0:7C 00 A0 7C A0 A0 A0 00<0F>
0FB8:7C 00 7C 7C 7C 7C 00 7C<19>
0FC0:00 7C 00 7C 7C 7C 00 7C<FF>
0FC8:7C 00 7C BA 30 B5 31 64<F2>
0FD0:00 A0 7C 00 A0 A0 7C 7C<34>
0FD8:A0 7C 7C A0 00 7C 64 64<98>
0FE0:7C 7C 7C A0 7C 7C 7C 7C<47>
0FE8:A0 7C 7C 7C 64 00 7C 7C<42>
0FF0:00 7C 7C 00 7C 00 7C 7C<0D>
0FF8:64 7C 7C 00 7C 00 7C 7C<47>
1000:00 7C 7C 64 64 7C 7C 7C<64>
1008:7C A0 7C 7C 00 7C 7C 7C<FA>
1010:7C 7C 7C 7C A0 7C 7C 7C<F6>
1018:7C 00 7C 00 7C 7C 00 7C<2F>
1020:7C 7C 00 7C 7C 00 7C 00<E9>
1028:00 7C 7C 7C A0 00 5F 30<A8>
1030:21 60 00 30 30 21 30 30<92>
1038:00 5F 00 30 30 30 30 00<D5>
1040:30 00 30 00 30 30 30 00<1D>
1048:30 30 00 30 90 00 30 30<1B>
1050:00 21 30 30 30 30 30 30<82>
1058:30 00 60 21 90 30 5F 30<58>
1060:30 30 30 30 5F 30 30 30<F1>
1068:30 5F 21 30 30 00 30 30<A9>
1070:00 30 00 30 30 5F 30 30<26>
1078:00 30 00 30 30 00 30 30<B1>
1080:21 60 30 5F 30 30 30 30<C7>
1088:5F 00 30 30 30 30 5F 30<80>
1090:30 60 30 30 30 30 00 60<88>
1098:00 30 60 00 60 30 30 00<B0>
10A0:60 30 00 60 00 00 60 60<13>
10A8:30 60 05 B1 BE 51 B1 00<39>
10B0:81 B1 51 BE B1 00 B1 00<4A>
10B8:81 B1 B1 B1 00 B1 00 B1<FA>
10C0:00 B1 B1 B1 00 B1 B1 00<62>
10C8:81 BE 00 B6 31 B1 32 81<25>
10D0:B1 00 51 BE B1 B1 B1 B1<BE>
10D8:B1 81 00 B1 51 BE 81 B1<02>
10E0:81 81 81 81 81 B1 81 81<BC>
10E8:81 B1 BE 51 B1 B1 00 B1<F3>
10F0:B1 00 81 00 81 B1 BE B1<18>
10F8:81 00 81 00 B1 B1 00 B1<EA>
1100:B1 51 BE B1 B1 81 81 81<E3>
1108:81 B1 00 81 81 81 81 B1<E7>
1110:81 B1 81 B1 B1 81 81 00<41>
1118:B1 00 81 B1 00 B1 81 81<04>
1120:00 B1 81 00 B1 00 00 B1<15>
1128:B1 B1 B1 2D 2D FF C8 C8<88>
1130:C8 C8 C8 FF 2D 2D 2D 2D<B6>
1138:C8 C8 C8 C8 C8 C8 C8 C8<5A>
1140:C8 C8 C8 C8 C8 C8 C8 C8<B6>
1148:C8 C8 C8 C8 C8 C8 C8 C8<8E>
1150:2D 2D 2D 2D 2D 2D 2D 2D<F1>
1158:2D 2D 2D 2D 2D 2D 2D 2D<D9>
1160:2D 2D 2D 2D 2D 2D 2D 2D<82>
1168:C8 C8 C8 C8 C8 C8 C8 C8<2F>
1170:C8 C8 C8 C8 C8 C8 C8 C8<27>
1178:C8 C8 C8 C8 C8 C8 C8 C8<1F>
1180:C8 C8 C8 C8 C8 C8 C8 C8<F7>
1188:C8 C8 C8 C8 C8 C8 C8 C8<CF>
1190:C8 C8 C8 C8 C8 C8 C8 C8<C6>
1198:C8 C8 C8 C8 C8 C8 C8 C8<BE>
11A0:C8 C8 C8 C8 C8 C8 C8 00<0F>
11A8:C8 C8 00 2D 2D 2D 2D 2D<18>
11B0:00 C8 C8 C8 C8 2D 2D 2D<D4>
11B8:2D 2D 2D 2D 2D 2D 2D 2D<BA>
11C0:2D 2D 2D 2D 2D 2D 2D 2D<A2>
11C8:2D 2D 2D B2 32 76 33 2D<17>
11D0:C8 C8 C8 C8 C8 C8 C8 C8<07>
11D8:C8 C8 C8 C8 C8 C8 C8 C8<FF>
11E0:C8 C8 C8 C8 C8 C8 C8 C8<96>
11E8:2D 2D 2D 2D 2D 2D 2D 2D<09>
11F0:2D 2D 2D 2D 2D 2D 2D 2D<11>
11F8:2D 2D 2D 2D 2D 2D 2D 2D<F9>
1200:2D 2D 2D 2D 2D 2D 2D 2D<DF>
1208:2D 2D 2D 2D 2D 2D 2D 2D<27>
1210:2D 2D 2D 2D 2D 2D 2D 2D<30>
1218:2D 2D 2D 2D 2D 2D 2D 2D<17>
1220:2D 2D 2D 2D 2D 2D 2D 42<51>
1228:24 81 3C 3C 3C 41 A4 00<14>
1230:00 00 3C 3C 3C 00 00 1A<93>
1238:06 1A 2E 42 56 6A 7E 92<AF>
1240:7E 6A 56 42 2E 1A 06 1A<8C>
1248:1D 09 1D 31 45 59 6D 81<BD>
1250:95 81 6D 59 45 31 1D 09<24>
1258:1D 20 0C 20 34 48 5C 70<FD>
1260:84 98 84 70 5C 48 34 20<FC>
1268:0C 20 17 03 17 2B 3F 53<E4>
1270:67 7B 8F 7B 67 53 3F 2B<EE>
1278:17 03 17 23 0F 23 37 4B<BC>
1280:5F 73 87 9B 87 73 5F 4B<C9>
1288:37 23 0F 23 30 0E 00 1F<FD>
1290:48 5C 72 00 77 33 35 34<5D>
1298:A2 00 8E 08 D2 A9 FF 8D<9C>
12A0:02 D2 A9 AF 8D 03 D2 A0<36>
12A8:02 20 A0 33 A9 F3 8D 02<2B>
12B0:D2 A0 02 20 A0 33 E8 E0<61>
12B8:0A D0 E2 A9 00 8D 00 D2<B4>
12C0:60 84 D4 A9 00 85 14 A5<AA>
12C8:14 C5 D4 D0 FA 60 F3 79<5B>
12D0:D9 6C C1 60 B6 5B A2 51<49>
12D8:90 48 80 40 A2 00 A9 05<36>
12E0:8D 08 D2 BD AD 33 8D 02<20>
12E8:D2 A9 AF 8D 03 D2 A0 08<F4>
12F0:20 A0 33 E8 E0 0E D0 EB<B8>
12F8:60 AD 7F 30 C9 01 D0 56<CA>
1300:EE 80 30 AD 80 30 C9 05<51>
1308:D0 08 A9 00 8D 80 30 EE<E1>
1310:81 30 AE 81 30 BD E5 2B<09>
1318:8D 04 D2 A9 AA 38 ED 80<F6>
1320:30 ED 80 30 8D 05 D2 EC<94>
1328:7E 30 D0 2A A9 00 8D 7F<93>
1330:30 8D 05 D2 AD 7E 30 18<37>
1338:69 08 8D 7E 30 AD 81 30<75>
1340:C9 19 90 0A A9 00 8D 81<83>
1348:30 A9 08 8D 7E 30 A2 0C<4E>
1350:20 87 40 20 B4 3C 60 36<6B>
1358:34 31 35 20 67 34 AE 6F<DB>
1360:30 BD EB 2F C9 00 F0 23<67>
1368:8D 02 D0 BC 95 31 88 A9<AD>
1370:FC A2 00 99 00 7E C8 E8<34>
1378:E0 03 D0 F7 AE 6F 30 BD<07>
1380:95 31 38 E9 06 A8 A9 30<85>
1388:99 00 7B 60 A2 00 A9 00<27>
1390:9D 00 7E 9D 00 7B E8 E0<48>
1398:FF D0 F3 60 AD 0E D0 C9<9A>
13A0:00 F0 05 A9 01 8D 69 30<3B>
13A8:AD 69 30 C9 01 F0 0E A9<7E>
13B0:B9 8D 06 D0 8D 68 30 A9<23>
13B8:06 8D C2 02 60 AD 69 30<30>
13C0:C9 01 F0 01 60 A2 00 CE<F5>
13C8:68 30 E8 E0 06 D0 F8 AD<B6>
13D0:0A D2 8D C2 02 AD 68 30<F1>
13D8:8D 06 D0 8D 04 D2 A9 86<D8>
13E0:8D 05 D2 AD 68 30 C9 14<BC>
13E8:B0 10 A9 00 8D 69 30 8D<95>
13F0:04 D2 8D 05 D2 A9 B9 8D<C2>
13F8:68 30 60 AD 1F D0 C9 07<E5>
1400:D0 05 A9 00 8D 78 30 AD<B6>
1408:78 30 C9 00 D0 28 AD 1F<29>
1410:D0 C9 06 D0 21 A9 40 8D<C8>
1418:0E D4 A2 00 8E 2F 02 8A<6F>
1420:9D 00 D0 E8 E0 0C D0 F8<E7>
1428:A9 01 8D 78 30 20 FD 38<4E>
1430:20 96 40 4C 72 49 60 AD<BF>
1438:04 D0 C9 05 D0 05 A9 01<A6>
```

Listing zu »Submission« (Fortsetzung)



```
1440:8D 6B 30 AD 05 D0 C9 04<17>
1448:D0 05 A9 01 8D 6B 30 C9<F6>
1450:05 D0 05 A9 01 8D 6B 32<5D>
1458:35 E5 35 30 60 AD 6B 30<B4>
1460:C9 01 F0 01 60 A9 00 85<E6>
1468:14 20 52 2D EE C0 02 CE<00>
1470:C1 02 AD C0 02 8D 00 D2<9F>
1478:A9 C8 8D 01 D2 A5 14 C9<43>
1480:64 D0 E9 A9 00 8D 00 D0<78>
1488:8D 01 D0 CE 7C 48 20 52<2D>
1490:2D 20 BB 33 20 77 33 AD<C0>
1498:7C 48 C9 10 D0 03 4C F0<FC>
14A0:34 20 52 2D 4C A6 49 AD<40>
14A8:0A D0 C9 02 D0 05 A9 01<E7>
14B0:8D 6B 30 C9 03 D0 05 A9<9A>
14B8:01 8D 6B 30 60 AD 86 30<E7>
14C0:C9 01 F0 22 AD 05 D0 C9<8F>
14C8:09 F0 04 C9 01 D0 17 AD<AF>
14D0:70 30 C9 64 90 06 CE 70<BC>
14D8:30 4C BE 35 C9 63 B0 06<71>
14E0:EE 70 30 4C BE 35 60 20<A5>
14E8:A9 3A AD 87 30 C9 00 F0<06>
14F0:05 A9 1F 8D 88 30 60 AE<E5>
14F8:6F 30 BD DD 4A C9 00 D0<AE>
1500:0D A2 19 20 87 40 AE 6F<3D>
1508:30 A9 01 9D DD 4A 60 E6<65>
1510:35 E1 36 EE 6D 30 AD 6D<A5>
1518:30 C9 14 D0 17 A9 00 8D<F8>
1520:6D 30 8D 6E 30 20 35 3F<C0>
1528:20 B7 3E 20 45 3F 20 89<CA>
1530:3F 20 DF 3F C9 0A D0 0B<C0>
1538:20 45 3F 20 89 3F A9 01<01>
1540:8D 6E 30 A9 CE 18 6D 6E<C2>
1548:30 8D 1B 23 8D 16 25 8D<9D>
1550:D9 27 60 AE 6F 30 A9 00<1D>
1558:8D 2F 02 A9 00 8D 30 02<84>
1560:A9 06 8D 31 02 BD 35 28<3B>
1568:85 58 BD B2 28 85 59 A0<D2>
1570:00 B1 58 99 00 06 C8 C0<D3>
1578:50 D0 F6 BD 9B 30 8D 03<43>
1580:D0 20 79 36 20 36 34 20<6F>
1588:CE 35 20 8D 37 20 2D 37<7D>
1590:A9 3E 8D 2F 02 60 A9 00<02>
1598:85 14 A9 00 8D 1E D0 A5<5E>
15A0:14 C9 14 D0 F5 60 A2 00<F2>
15A8:A9 00 9D 00 7F E8 E0 FF<35>
15B0:D0 F6 AE 6F 30 BD 18 31<E2>
15B8:8D 93 30 BD 95 31 8D 94<C6>
15C0:30 AE 93 30 A9 C3 9D 00<11>
15C8:7F E8 EC 94 30 D0 F5 AE<3F>
15D0:93 30 A9 FF 9D 00 7F 9D<D6>
15D8:01 7F 8A 18 69 0A AA EC<E1>
15E0:94 30 90 EE 60 A2 00 AC<8A>
15E8:71 30 BD F4 2D 99 00 7C<F5>
15F0:BD 18 2E 99 00 7D C8 E8<A9>
15F8:E0 24 D0 EE 60 A2 00 AC<29>
1600:71 30 BD 64 2D 99 00 7C<12>
1608:BD 88 2D 99 00 7D C8 E2<47>
1610:36 47 37 E8 E0 24 D0 EE<BD>
1618:60 A2 00 AC 71 30 BD 3C<91>
1620:2E 99 00 7C BD 60 2E 99<A2>
1628:00 7D C8 E8 E0 23 D0 EE<44>
1630:60 A2 00 AC 71 30 BD AC<E9>
1638:2D 99 00 7C BD D0 2D 99<09>
1640:00 7D C8 E8 E0 23 D0 EE<1C>
1648:60 EE 72 30 AD 72 30 C9<1F>
1650:02 D0 05 A9 00 8D 72 30<F7>
1658:A9 00 8D 73 30 60 EE 77<63>
1660:30 AD 77 30 C9 02 D0 0B<B0>
1668:A9 A6 8D C6 02 A9 00 8D<04>
1670:77 30 60 A9 36 8D C6 02<80>
1678:60 48 37 43 38 A2 00 9D<D7>
1680:F8 80 E8 E0 08 D0 F8 60<52>
1688:A2 00 9D 00 81 E8 E0 08<AC>
1690:D0 F8 60 EE 83 30 AD 83<40>
1698:30 C9 32 D0 0A A9 18 20<95>
16A0:48 37 A9 00 20 53 37 C9<59>
16A8:64 D0 0A A9 00 20 48 37<CE>
16B0:A9 18 20 53 37 C9 96 D0<38>
16B8:08 A9 00 8D 83 30 20 53<54>
16C0:37 60 20 4B 38 AE 6F 30<5E>
16C8:BD DD 4A C9 01 D0 13 AD<7E>
16D0:6A 30 8D 89 25 8D 90 25<73>
16D8:38 E9 01 8D 66 23 8D 6D<89>
16E0:23 60 C9 02 D0 15 A9 00<13>
16E8:8D 89 25 8D 66 23 AD 6A<1F>
16F0:30 8D 90 25 38 E9 01 8D<90>
16F8:6D 23 60 C9 03 D0 15 A9<6A>
1700:00 8D 90 25 8D 6D 23 AD<A2>
1708:6A 30 8D 89 25 38 E9 01<6E>
1710:8D 66 23 60 C9 04 D0 0E<74>
1718:A9 00 8D 89 25 8D 90 25<56>
1720:8D 66 23 8D 6D 23 60 AD<CF>
1728:05 D0 C9 08 D0 34 A9 01<08>
1730:8D 7F 30 AD 70 30 C9 78<5A>
1738:90 11 A9 00 8D 90 25 8D<01>
1740:6D 23 AE 6F 30 A9 03 9D<E1>
1748:DD 4A 60 AD 70 30 C9 6E<7A>
1750:B0 10 A9 00 8D 89 25 8D<CB>
1758:66 23 AE 6F 30 A9 02 9D<D0>
1760:DD 4A AD 90 25 C9 00 D0<8A>
1768:10 AD 90 25 CD 66 23 D0<44>
1770:08 A9 04 AE 6F 30 9D DD<78>
1778:4A 44 38 CC 38 AD EB 4A<14>
1780:8D FB 4A 60 AE 75 30 BD<D4>
1788:BE 2F 8D 6A 30 60 EE 76<0A>
1790:30 AD 76 30 C9 05 D0 0D<DC>
1798:A2 00 BD 0A 33 9D 10 81<78>
17A0:E8 E0 08 D0 F5 C9 0A D0<AB>
17A8:10 A2 00 8E 76 30 BD 12<1C>
17B0:33 9D 10 81 E8 E0 08 D0<38>
17B8:F5 60 EE 82 30 AD 82 30<C2>
17C0:C9 08 D0 1C A9 9C 8D 27<FE>
17C8:26 8D 2E 26 8D FE 27 8D<4C>
17D0:06 28 A9 9D 8D 2A 26 8D<68>
17D8:31 26 8D FF 27 8D 07 28<6B>
17E0:C9 10 D0 21 A9 9D 8D 27<00>
17E8:26 8D 2E 26 8D FE 27 8D<AC>
17F0:06 28 A9 9C 8D 2A 26 8D<F8>
17F8:31 26 8D FF 27 8D 07 28<4B>
1800:A9 00 8D 82 30 60 CD 38<2E>
1808:C8 39 48 18 AD 0B D4 C9<F7>
1810:46 B0 10 AD 74 30 8D 0A<B1>
1818:D4 8D 1A D0 A9 80 8D 09<FC>
1820:D4 68 40 A9 78 8D 0A D4<7C>
1828:8D 09 D4 A9 00 8D 18 D0<38>
1830:8D 1A D0 A9 1A 8D 17 D0<B2>
1838:68 40 A2 00 A9 00 9D 00<E6>
1840:7C 9D 00 7D 9D 00 7E 9D<AC>
1848:00 7B E8 E0 FF D0 ED 60<83>
1850:EE 9A 30 AD 9A 30 C9 03<3D>
1858:D0 49 20 67 39 20 A0 39<40>
1860:20 D9 39 20 12 3A A9 00<C2>
1868:8D 9A 30 EE 95 30 AC 95<02>
1870:30 BE 1B 33 A9 A7 18 6D<3F>
1878:8B 30 9D 37 26 9D E1 26<FC>
1880:C8 BE 1B 33 A9 00 9D 37<58>
1888:26 9D E1 26 88 88 BE 1B<9C>
1890:33 9D 37 26 9D E1 26 AD<34>
1898:95 30 C9 0F D0 05 A9 00<15>
18A0:8D 95 30 60 EE 96 30 AC<42>
18A8:96 30 BE 2C 33 A9 A7 18<CD>
18B0:6D 8B 30 9D 37 26 9D E1<34>
18B8:26 C8 BE 2C 33 A9 00 9D<41>
18C0:37 26 9D E1 26 88 88 BE<51>
18C8:2C 33 9D 37 26 9D E1 26<DD>
18D0:AD 96 30 C9 0F D0 05 A9<69>
18D8:00 8D 96 30 60 EE 97 30<4D>
18E0:AC 97 30 BE 3D 33 A9 A7<A9>
18E8:18 6D 8B 30 9D 37 26 9D<7E>
18F0:E1 26 C8 BE 3D 33 A9 00<FE>
18F8:9D 37 26 9D E1 26 88 88<4B>
1900:BE 3D 33 9D 37 26 C9 39<AD>
1908:4A 3A 9D E1 26 AD 97 30<DC>
1910:C9 0F D0 05 A9 00 8D 97<AA>
1918:30 60 EE 98 30 AC 98 30<41>
1920:BE 4E 33 A9 A7 18 6D 8B<70>
1928:30 9D 37 26 9D E1 26 C8<47>
1930:BE 4E 33 A9 00 9D 37 26<CC>
1938:9D E1 26 88 88 BE 4E 33<3F>
1940:9D 37 26 9D E1 26 AD 98<BE>
1948:30 C9 0F D0 05 A9 00 8D<7A>
1950:98 30 60 EE 99 30 AC 99<1B>
1958:30 BE 5F 33 A9 A7 18 6D<A6>
1960:8B 30 9D 37 26 9D E1 26<1A>
1968:C8 BE 5F 33 A9 00 9D 37<43>
1970:26 9D E1 26 88 88 BE 5F<98>
1978:33 9D 37 26 9D E1 26 AD<56>
1980:99 30 C9 0F D0 05 A9 00<39>
1988:8D 99 30 60 4B 3A 46 3B<6E>
1990:EE 73 30 AD 73 30 C9 08<15>
1998:D0 03 20 18 37 A9 00 8D<E2>
19A0:85 30 60 AD 86 30 C9 00<76>
19A8:D0 43 AD 78 02 C9 07 D0<B5>
19B0:18 EE 70 30 20 A9 3A AD<44>
19B8:72 30 C9 00 D0 08 A9 A4<64>
19C0:8D 6C 30 4C D0 36 4C 00<CE>
19C8:37 C9 0B D0 18 CE 70 30<AF>
19D0:20 A9 3A AD 72 30 C9 00<5C>
19D8:D0 08 A9 A4 8D 6C 30 4C<84>
19E0:B8 36 4C E8 36 AD 87 30<87>
19E8:C9 01 D0 01 EA 60 AD 70<E5>
19F0:30 8D 00 D0 8D 01 D0 AE<18>
19F8:6F 30 DD 12 32 D0 09 EE<7F>
1A00:6F 30 20 26 36 4C 0B 3C<8B>
1A08:DD 8E 32 D0 09 CE 6F 30<DF>
1A10:20 26 36 4C 0B 3C 60 60<B3>
1A18:20 77 34 AD 0F D0 C9 00<C4>
1A20:F0 50 20 7B 3B AD 78 02<9B>
1A28:C9 0D D0 12 AD 85 30 C9<E4>
1A30:01 F0 0B EE 71 30 A9 00<BE>
1A38:8D 85 30 20 33 3B 20 C6<68>
1A40:3B AD 78 02 C9 0E D0 12<3C>
1A48:AD 85 30 C9 02 F0 0B CE<7A>
1A50:71 30 A9 00 8D 85 30 20<35>
1A58:33 3B AD 85 30 C9 00 D0<4B>
1A60:11 AE 6F 30 BD 9B 30 8D<BD>
1A68:70 30 20 A9 3A A9 00 8D<7C>
1A70:85 30 A9 00 8D 1E D0 60<6D>
1A78:AD 72 30 C9 00 D0 03 4C<A3>
1A80:41 3B 4C 5E 3B 60 A2 00<22>
1A88:AC 71 30 BD 47 3B 42 3C<A6>
1A90:26 2F 99 00 7C BD 4A 2F<67>
1A98:99 00 7D E8 C8 E0 24 D0<FD>
1AA0:EE A9 01 8D 86 30 60 A2<7A>
1AA8:00 AC 71 30 BD 6E 2F 99<93>
1AB0:00 7C BD 92 2F 99 00 7D<ED>
1AB8:C8 E8 E0 24 D0 EE A9 01<99>
1AC0:8D 86 30 60 AE 6F 30 AD<C7>
1AC8:71 30 18 69 21 DD 95 31<71>
1AD0:D0 3C A9 01 8D 85 30 A9<2C>
1AD8:00 8D 86 30 AE 6F 30 BD<6B>
1AE0:95 31 C9 BE D0 28 EE 6F<92>
1AE8:30 AD 6F 30 C9 3D F0 0A<4D>
1AF0:C9 09 90 06 CE 6F 30 CE<93>
1AF8:6F 30 20 FD 38 20 26 36<88>
1B00:AE 6F 30 BD 18 31 69 01<10>
1B08:8D 71 30 20 41 3B 60 AD<4B>
1B10:71 30 AE 6F 30 DD 18 31<C5>
1B18:D0 39 A9 02 8D 85 30 A9<21>
1B20:00 8D 86 30 AE 6F 30 BD<B1>
1B28:18 31 C9 21 D0 25 CE 6F<EF>
1B30:30 AD 6F 30 C9 3C F0 0A<BB>
1B38:C9 07 90 06 EE 6F 30 EE<5E>
1B40:6F 30 20 26 36 BD 95 31<10>
1B48:E9 23 8D 71 30 20 FD 38<88>
1B50:20 5E 3B 60 18 AD 70 30<23>
1B58:69 96 8D 70 30 90 05 A9<04>
1B60:30 8D 70 30 AD 70 30 8D<E5>
1B68:00 D0 8D 01 D0 60 AD 88<13>
1B70:30 6D 88 30 8D 00 D2 A9<80>
1B78:A4 8D 01 D2 60 A9 28 8D<A8>
1B80:00 D2 AD 89 30 8D 01 D2<E1>
1B88:60 AD 6C 30 43 3C 39 3D<43>
1B90:C9 A0 F0 11 CE 6C 30 CE<70>
1B98:6C 30 A9 C8 8D 06 D2 AD<F7>
1BA0:6C 30 8D 07 D2 60 AD 05<09>
1BA8:D0 C9 02 D0 32 20 52 2D<52>
1BB0:20 85 2C A0 00 A2 FA 20<32>
1BB8:87 40 C8 C0 0A D0 F6 AD<3F>
1BC0:7C 48 C9 15 F0 03 EE 7C<E4>
1BC8:48 AD 7D 30 C9 00 F0 F9<97>
1BD0:20 44 2D 20 C7 3C EE 75<46>
1BD8:30 EE 72 48 4C A6 49 60<4F>
1BE0:AE 75 30 E0 07 D0 0B A9<D5>
1BE8:00 8D 75 30 AA A9 11 8D<85>
1BF0:72 48 BD 6F 33 8D 6F 30<D4>
1BF8:BC 6F 33 A9 01 99 DD 4A<4B>
1C00:60 EE 6C 48 AD 6C 48 C9<62>
1C08:1A D0 08 EE 6B 48 A9 10<FE>
1C10:8D 6C 48 60 EA CE 6C 48<62>
1C18:AD 6C 48 C9 0F D0 0F AD<99>
1C20:6B 48 C9 10 F0 37 CE 6B<1E>
1C28:48 A9 19 8D 6C 48 A2 FF<31>
1C30:A9 00 85 14 8A 38 E9 0C<99>
1C38:AA A5 14 C9 01 D0 FA 8E<97>
1C40:00 D2 8A 18 69 01 8D 02<22>
1C48:D2 A9 A8 8D 01 D2 8D 03<3D>
1C50:D2 E0 0A B0 DB A2 32 20<0B>
1C58:87 40 4C C7 3C A9 00 8D<6F>
1C60:00 D2 8D 01 D2 8D 02 D2<69>
1C68:8D 03 D2 60 AD 0A D2 30<D1>
1C70:0B A9 00 8D D8 30 A9 64<5B>
1C78:8D EE 30 60 A9 00 8D EE<4D>
1C80:30 A9 64 8D D8 30 60 3A<F0>
1C88:3D 35 3E AD 89 30 C9 00<E3>
1C90:D0 3C AD 87 30 C9 00 D0<42>
1C98:35 AD 86 30 C9 00 D0 2E<5E>
1CA0:AD 84 02 C9 00 D0 27 AD<27>
1CA8:78 02 C9 0B D0 08 A9 01<D7>
1CB0:8D 87 30 4C 83 3D C9 07<41>
1CB8:D0 08 A9 02 8D 87 30 4C<36>
1CC0:83 3D C9 0F D0 08 A9 03<46>
```

Listing zu »Submission« (Fortsetzung)

```
1CC8:8D 87 30 4C 83 3D A9 00<C3>
1CD0:8D 87 30 60 A9 00 8D 88<A5>
1CD8:30 CE 71 30 CE 71 30 CE<0B>
1CE0:71 30 CE 71 30 60 AD 87<8D>
1CE8:30 C9 01 D0 06 CE 70 30<FF>
1CF0:4C A9 3D C9 02 D0 03 EE<20>
1CF8:70 30 20 A9 3A 20 9E 3E<CE>
1D00:AD 87 30 C9 01 D0 2E AD<55>
1D08:88 30 C9 10 B0 03 4C 84<D0>
1D10:3E C9 1E B0 03 4C CA 3D<74>
1D18:4C CA 3D AE 71 30 A0 00<01>
1D20:B9 84 2E 9D 00 7C B9 AE<34>
1D28:2E 9D 00 7D E8 C8 C0 24<82>
1D30:D0 EE 4C 12 3E AD 88 30<5C>
1D38:C9 10 B0 03 4C 6A 3E C9<4E>
1D40:1E B0 03 4C F8 3D 4C F8<9F>
1D48:3D AE 71 30 A0 00 B9 D2<FE>
1D50:2E 9D 00 7C B9 FD 2E 9D<3B>
1D58:00 7D E8 C8 C0 24 D0 EE<87>
1D60:4C 12 3E EE 88 30 20 25<4E>
1D68:3C 20 94 35 AD 88 30 C9<5F>
1D70:1E 90 36 A9 00 8D 00 D2<E3>
1D78:8D 1E D0 A9 10 8D 89 30<5B>
1D80:EE 71 30 EE 71 30 EE 36<10>
1D88:3E B6 3E 71 30 EE 71 30<78>
1D90:AD 87 30 C9 01 D0 0B A9<5B>
1D98:00 8D 87 30 20 9E 3E 4C<49>
1DA0:E8 36 A9 00 8D 87 30 20<7A>
1DA8:9E 3E 4C 00 37 60 AD 89<A8>
1DB0:30 C9 00 F0 09 CE 89 30<B8>
1DB8:CE 89 30 20 34 3C 60 AE<25>
1DC0:71 30 A0 1D B9 84 2E 9D<D2>
1DC8:00 7C B9 AE 2E 9D 00 7D<98>
1DD0:E8 88 C0 00 D0 EE 4C 12<61>
1DD8:3E AE 71 30 A0 1D B9 D2<E5>
1DE0:2E 9D 00 7C B9 FD 2E 9D<C9>
1DE8:00 7D E8 88 C0 00 D0 EE<62>
1DF0:4C 12 3E AE 71 30 CA CA<19>
1DF8:CA CA CA A0 00 A9 00 9D<10>
1E00:00 7C 9D 00 7D E8 C8 C0<83>
1E08:32 D0 F4 60 B7 3E B2 3F<14>
1E10:AD 8C 30 C9 18 D0 05 A9<CB>
1E18:00 8D 8A 30 C9 21 D0 05<5B>
1E20:A9 01 8D 8A 30 AD 8A 30<A6>
1E28:C9 00 F0 31 CE 8C 30 AE<88>
1E30:8C 30 A9 90 9D 34 22 A9<FF>
1E38:91 9D 35 22 A9 92 9D 36<A9>
1E40:22 A9 00 9D 37 22 AD 8B<67>
1E48:30 C9 01 D0 0F A9 93 9D<CD>
1E50:34 22 A9 95 9D 36 22 A9<48>
1E58:94 9D 35 22 60 EE 8C 30<C8>
1E60:AE 8C 30 A9 96 9D 34 22<14>
1E68:A9 94 9D 35 22 A9 99 9D<C7>
1E70:36 22 A9 00 9D 33 22 AD<08>
1E78:8B 30 C9 00 D0 0F A9 97<A2>
1E80:9D 34 22 A9 98 9D 36 22<A4>
1E88:A9 91 9D 35 22 60 EE 8B<5D>
1E90:30 AD 8B 30 C9 02 D0 05<57>
1E98:A9 00 8D 8B 30 60 AD 8E<6F>
1EA0:30 C9 01 D0 03 CE 8D 30<E0>
1EA8:C9 00 D0 03 EE 8D 30 AD<22>
1EB0:8D 30 C9 1A D0 05 A9 00<DF>
1EB8:8D 8E 30 C9 23 D0 05 A9<1C>
1EC0:01 8D 8E 30 AE 8D 30 A9<1E>
1EC8:9A 18 6D 8B 30 9D 74 23<9B>
1ED0:9D 43 25 A9 00 9D 44 25<0A>
1ED8:9D 42 25 9D 75 23 9D 73<C8>
1EE0:23 60 AD 90 30 C9 01 D0<AA>
1EE8:03 CE 8F 30 C9 00 D0 03<D8>
1EF0:EE 8F 30 AD 8F 30 C9 18<76>
1EF8:D0 05 A9 00 8D 90 30 C9<46>
1F00:21 D0 05 A9 01 8D 90 30<89>
1F08:AE 8F 30 A9 B3 3F 37 40<98>
1F10:9C 18 6D 8B 30 9D 14 24<B8>
1F18:9D 0D 25 9D AC 22 A9 9D<28>
1F20:38 ED 8B 30 9D AD 22 A9<96>
1F28:00 9D 0C 25 9D 0E 25 9D<5D>
1F30:15 24 9D 13 24 9D AB 22<B3>
1F38:9D AE 22 60 AD 92 30 C9<93>
1F40:01 D0 03 CE 91 30 C9 00<0F>
1F48:D0 03 EE 91 30 AD 91 30<FD>
1F50:C9 18 D0 05 A9 00 8D 92<B4>
1F58:30 C9 21 D0 05 A9 01 8D<5B>
1F60:92 30 AE 91 30 A9 90 18<13>
1F68:6D 8B 30 6D 8B 30 6D 8B<55>
1F70:30 9D 84 22 A9 91 18 6D<FD>
1F78:8B 30 6D 8B 30 6D 8B 30<22>
1F80:9D 85 22 A9 98 18 6D 8B<D0>
1F88:30 9D 86 22 A9 00 9D 87<08>
1F90:22 9D 83 22 60 38 40 E5<E2>
1F98:40 EE 62 48 AD 62 48 C9<7B>
1FA0:1A D0 08 EE 61 48 A9 10<35>
1FA8:8D 62 48 AD 61 48 C9 1A<31>
1FB0:D0 08 EE 60 48 A9 10 8D<2E>
1FB8:61 48 AD 60 48 C9 1A D0<2E>
1FC0:08 EE 5F 48 A9 10 8D 60<8B>
1FC8:48 AD 5F 48 C9 1A D0 08<F9>
1FD0:EE 5E 48 A9 10 8D 5F 48<CD>
1FD8:AD 5E 48 C9 1A D0 08 EE<E7>
1FE0:5D 48 A9 10 8D 5E 48 60<C3>
1FE8:8E 84 30 A2 00 20 38 40<AF>
1FF0:E8 EC 84 30 D0 F7 60 A2<45>
1FF8:FF E8 BD 5D 48 C9 10 F0<92>
2000:F8 8E 93 30 A2 FF E8 BD<8A>
2008:EF 29 C9 10 F0 F8 8E 94<F2>
2010:30 AD 93 30 18 CD 94 30<34>
2018:B0 0F AE 93 30 BD 5D 48<4C>
2020:9D EF 29 E8 E0 06 D0 F5<B9>
2028:60 D0 1B AE 93 30 BD EF<10>
2030:29 18 DD 5D 48 90 E3 D0<12>
2038:0D EE 93 30 AD 93 30 C9<AE>
2040:08 F0 03 4C CA 40 60 E6<97>
2048:40 E1 41 70 70 C6 0C 22<D4>
2050:46 5C 22 06 46 EC 23 06<55>
2058:46 EC 23 06 46 5C 22 06<69>
2060:46 D4 22 06 46 FC 22 06<08>
2068:46 5C 22 06 46 0C 22 06<2A>
2070:46 34 22 06 46 5C 22 86<EF>
2078:42 2E 48 02 02 41 E6 40<AF>
2080:70 70 C6 0C 22 46 5C 22<4F>
2088:06 46 0C 22 06 46 34 22<C4>
2090:06 46 5C 22 06 46 0D 28<9A>
2098:06 46 0D 28 06 46 9C 23<25>
20A0:06 46 4C 23 06 46 74 23<76>
20A8:06 46 5C 22 86 42 2E 48<E8>
20B0:02 02 41 1B 41 70 70 C6<BA>
20B8:0C 22 46 5C 22 06 46 4C<D7>
20C0:23 06 46 74 23 06 46 9C<6C>
20C8:23 06 46 9C 23 06 46 9C<E7>
20D0:23 06 46 9C 23 06 46 9C<FF>
20D8:23 06 46 9C 23 06 46 9C<17>
20E0:23 86 42 2E 48 02 02 41<C5>
20E8:50 41 70 70 C6 0C 22 46<8A>
20F0:9C 23 06 46 24 23 06 46<2B>
20F8:24 23 06 46 5C 22 06 46<F5>
2100:0D 28 06 46 0D 28 06 46<4B>
2108:9C 23 06 46 24 23 06 46<05>
2110:74 23 06 46 5C 22 86 42<B2>
2118:2E 48 02 02 41 85 41 70<B4>
2120:70 C6 0C 22 46 5C 22 06<15>
2128:46 31 27 06 06 06 46 5C<CF>
2130:22 06 46 31 27 06 06 06<B3>
2138:46 5C 22 06 46 31 27 06<22>
2140:06 06 46 5C 22 86 42 E2<13>
2148:41 DD 42 2E 48 02 02 41<1C>
2150:BA 41 70 70 C6 0C 22 46<ED>
2158:5C 22 06 46 C4 23 06 46<73>
2160:C4 23 06 46 7F 24 06 46<40>
2168:EC 23 06 46 14 24 06 46<11>
2170:5C 22 06 46 0C 22 06 46<C9>
2178:34 22 06 46 5C 22 86 42<2A>
2180:2E 48 02 02 41 E9 41 70<EA>
2188:70 C6 0C 22 46 5C 22 06<BD>
2190:46 4C 23 06 46 74 23 06<D9>
2198:46 9C 23 06 46 9C 23 06<96>
21A0:46 9C 23 06 46 9C 23 06<5E>
21A8:46 EC 23 06 46 14 24 06<1C>
21B0:46 5C 22 86 42 2E 48 02<B2>
21B8:02 41 1E 42 70 70 C6 0C<BC>
21C0:22 46 9C 23 06 46 EC 23<AE>
21C8:06 46 14 24 06 46 5C 22<63>
21D0:06 46 5C 22 06 46 5C 22<76>
21D8:06 46 5C 22 06 46 5C 22<7E>
21E0:06 46 5C 22 06 46 5C 22<66>
21E8:86 42 2E 48 02 02 41 53<15>
21F0:42 70 70 C6 0C 22 46 5C<AB>
21F8:22 06 46 C4 23 06 46 C4<53>
2200:23 06 46 9C 23 06 46 9C<23>
2208:23 06 46 1F 25 06 06 06<4D>
2210:46 9C 23 06 46 9C 23 06<F3>
2218:46 9C 23 86 42 2E 48 02<7E>
2220:02 41 88 42 70 70 C6 0C<A5>
2228:22 46 5C 22 06 46 24 23<2F>
2230:06 46 24 23 06 46 7F 24<E7>
2238:06 46 7F 24 06 46 7F 24<4B>
2240:06 46 7F 24 06 46 EC DE<26>
2248:42 D9 43 23 06 46 14 24<13>
2250:06 46 5C 22 86 42 2E 48<AC>
2258:02 02 41 BB 42 70 70 C6<46>
2260:3C 24 46 5C 22 06 46 24<32>
2268:23 06 46 24 23 06 46 5C<C2>
2270:22 06 46 6F 25 06 06 06<52>
2278:46 5C 22 06 46 D4 22 06<67>
2280:46 FC 22 06 46 5C 22 86<F9>
2288:42 2E 48 02 02 41 F0 42<B9>
2290:70 70 C6 0C 22 46 5C 22<23>
2298:06 46 EC 23 06 46 EC 23<55>
22A0:06 46 7F 24 06 46 7F 24<F3>
22A8:06 46 7F 24 06 46 7F 24<DB>
22B0:06 46 4C 23 06 46 74 23<5A>
22B8:06 46 5C 22 86 42 2E 48<D4>
22C0:02 02 41 23 43 70 70 C6<58>
22C8:0C 22 46 5C 22 06 46 5C<1A>
22D0:22 06 46 5C 22 06 46 5C<36>
22D8:22 06 46 5C 22 06 46 37<29>
22E0:26 06 06 06 06 06 06 06<34>
22E8:46 5C 22 86 42 2E 48 02<5E>
22F0:02 41 58 43 70 70 C6 0C<59>
22F8:22 46 5C 22 06 46 0C 22<EF>
2300:06 46 AC 22 06 46 5C 22<94>
2308:06 46 5C 22 06 46 5C 22<A2>
2310:06 46 5C 22 06 46 5C 22<BA>
2318:06 46 5C 22 06 46 5C 22<B2>
2320:86 42 2E 48 02 02 41 87<15>
2328:43 70 70 C6 0C 22 46 5C<E7>
2330:22 06 46 6F 25 06 06 06<9B>
2338:46 5C 22 06 46 5C 22 06<C6>
2340:46 5C 22 06 46 5C 22 DA<E3>
2348:43 D5 44 06 46 A7 24 06<65>
2350:46 A7 24 06 46 5C 22 86<B4>
2358:42 2E 48 02 02 41 BC 43<03>
2360:70 70 C6 0C 22 46 5C 22<79>
2368:06 46 5C 22 06 46 5C 22<C2>
2370:06 46 5C 22 06 46 24 23<6D>
2378:06 46 24 23 06 46 5C 22<2C>
2380:06 46 CF 24 06 46 F7 24<CE>
2388:06 46 5C 22 86 42 2E 48<E2>
2390:02 02 41 EF 43 70 70 C6<D8>
2398:0C 22 46 5C 22 06 46 24<D5>
23A0:23 06 46 74 23 06 46 5C<03>
23A8:22 06 46 EC 23 06 46 14<98>
23B0:24 06 46 5C 22 06 46 5C<A8>
23B8:22 06 46 5C 22 06 46 5C<A9>
23C0:22 86 42 2E 48 02 02 41<53>
23C8:24 44 70 70 C6 0C 22 46<1E>
23D0:5C 22 06 46 5C 22 06 46<9F>
23D8:E1 26 06 06 06 06 06 06<93>
23E0:06 46 5C 22 06 46 5C 22<6A>
23E8:06 46 5C 22 06 46 5C 22<42>
23F0:86 42 2E 48 02 02 41 59<F7>
23F8:44 70 70 C6 0C 22 46 5C<96>
2400:22 06 46 4C 23 06 46 74<78>
2408:23 06 46 9C 23 06 46 1F<C2>
2410:25 06 46 5C 22 06 46 9C<4C>
2418:23 06 46 EC 23 06 46 14<CC>
2420:24 06 46 5C 22 86 42 2E<42>
2428:48 02 02 41 88 44 70 70<8F>
2430:C6 0C 22 46 5C 22 06 46<9C>
2438:0C 22 06 46 84 22 06 46<1A>
2440:5C 22 06 46 5C 22 06 D6<A1>
2448:44 D1 45 46 1F 25 06 46<C3>
2450:5C 22 06 46 6F 25 06 06<92>
2458:06 46 5C 22 86 42 2E 48<A2>
2460:02 02 41 BD 44 70 70 C6<B0>
2468:0C 22 46 5C 22 06 46 C4<27>
2470:23 06 46 C4 23 06 46 5C<EA>
2478:22 06 46 6F 25 06 06 06<56>
2480:46 5C 22 06 46 D4 22 06<2B>
2488:46 FC 22 06 46 5C 22 86<FD>
2490:42 2E 48 02 02 41 F0 44<A7>
2498:70 70 C6 0C 22 46 5C 22<2F>
24A0:06 46 0C 22 06 46 E5 27<4B>
24A8:06 46 5C 22 06 46 6F 25<3C>
24B0:06 06 06 46 5C 22 06 46<4E>
24B8:0C 22 06 46 84 22 06 46<9E>
24C0:5C 22 86 42 2E 48 02 02<1C>
24C8:41 23 45 70 70 C6 0C 22<E1>
24D0:46 5C 22 06 46 0C 22 06<5B>
24D8:46 84 22 06 46 5C 22 06<F2>
24E0:46 6F 25 06 06 06 46 5C<5D>
24E8:22 06 46 0C 22 06 46 E5<E5>
24F0:27 06 46 5C 22 86 42 2E<B9>
24F8:48 02 02 41 56 45 70 70<4D>
2500:C6 0C 22 46 5C 22 06 46<92>
2508:6F 25 06 06 06 46 5C 22<17>
2510:06 46 0C 22 06 46 AC 22<55>
2518:06 46 5C 22 06 46 BF 25<70>
2520:06 46 BF 25 06 46 5C 22<4D>
2528:86 42 2E 48 02 02 41 89<F1>
2530:45 70 70 C6 0C 22 46 5C<D2>
2538:22 06 46 4C 23 06 46 24<72>
2540:23 06 46 5C 22 06 46 D2<93>
2548:45 CD 46 EC 23 06 46 14<C9>
```

Listing zu »Submission« (Fortsetzung)

```
2550:24 06 46 5C 22 06 46 0C<7C>
2558:22 06 46 34 22 06 46 5C<42>
2560:22 86 42 2E 48 02 02 41<B7>
2568:BC 45 70 70 C6 0C 22 46<CF>
2570:5C 22 06 46 EC 23 06 46<D4>
2578:0F 26 06 46 5C 22 06 46<B2>
2580:4C 23 06 46 24 23 06 46<DA>
2588:5C 22 06 46 0C 22 06 46<69>
2590:34 22 06 46 5C 22 86 42<4B>
2598:2E 48 02 02 41 F1 45 70<F2>
25A0:70 C6 0C 22 46 5C 22 06<9D>
25A8:46 0D 28 06 46 0D 28 06<F2>
25B0:46 5C 22 06 46 5C 22 06<43>
25B8:46 5C 22 06 46 5C 22 06<AA>
25C0:46 4C 23 06 46 74 23 06<11>
25C8:46 5C 22 86 42 2E 48 02<14>
25D0:02 41 26 46 70 70 C6 0C<69>
25D8:22 46 5C 22 06 46 E7 25<6E>
25E0:06 46 E7 25 06 46 5C 22<11>
25E8:06 46 CF 24 06 46 F7 24<FD>
25F0:06 46 5C 22 06 46 1F 25<E8>
25F8:06 46 5C 22 06 46 5C 22<86>
2600:86 42 2E 48 02 02 41 5B<BD>
2608:46 70 70 C6 0C 22 46 5C<7F>
2610:22 06 46 EC 23 06 46 EC<1E>
2618:23 06 46 5C 22 06 46 4C<E7>
2620:23 06 46 74 23 06 46 5C<81>
2628:22 06 46 5C 22 06 46 5C<16>
2630:22 06 46 5C 22 86 42 2E<F6>
2638:48 02 02 41 90 46 70 70<63>
2640:C6 0C 22 46 5C 22 06 CE<E1>
2648:46 C9 47 46 0C 22 06 46<6E>
2650:34 22 06 46 5C 22 06 46<11>
2658:D4 22 06 06 06 46 5C 22<5A>
2660:06 46 4C 23 06 46 24 23<71>
2668:06 46 5C 22 86 42 2E 48<8E>
2670:02 02 41 C5 46 70 70 C6<2D>
2678:0C 22 46 5C 22 06 46 5C<E3>
2680:22 06 46 5C 22 06 46 5C<DF>
2688:22 06 46 C4 23 06 46 C4<FC>
2690:23 06 46 5C 22 06 46 EC<1E>
2698:23 06 46 14 24 06 46 5C<03>
26A0:22 86 42 2E 48 02 02 41<F9>
26A8:F8 46 70 70 C6 0C 22 46<3F>
26B0:5C 22 06 46 E7 25 06 46<7A>
26B8:E7 25 06 46 5C 22 06 46<DB>
26C0:4C 23 06 46 74 23 06 46<8E>
26C8:5C 22 06 46 EC 23 06 46<2D>
26D0:14 24 06 46 5C 22 86 42<FC>
26D8:2E 48 02 02 41 2D 47 70<26>
26E0:70 C6 0C 22 46 5C 22 06<63>
26E8:46 95 27 06 06 06 46 5C<12>
26F0:22 06 46 EC 23 06 46 0F<21>
26F8:26 06 46 5C 22 06 46 BF<B3>
2700:25 06 46 BF 25 06 46 5C<38>
2708:22 86 42 2E 48 02 02 41<83>
2710:62 47 70 70 C6 0C 22 46<CE>
2718:5C 22 06 46 E7 25 06 46<CF>
2720:E7 25 06 46 5C 22 06 46<FD>
2728:5C 22 06 46 E1 26 06 06<73>
2730:06 06 06 06 06 46 5C 22<C3>
2738:86 42 2E 48 02 02 41 95<B4>
2740:47 70 70 C6 0C 22 46 CA<B7>
2748:47 C5 48 5C 22 06 46 0C<FB>
2750:22 06 46 AC 22 06 46 5C<C4>
2758:22 06 46 BF 25 06 46 BF<6A>
2760:25 06 46 5C 22 06 46 5C<52>
2768:22 06 46 5C 22 06 46 5C<E1>
2770:22 86 42 2E 48 02 02 41<2B>
2778:C4 47 70 70 C6 0C 22 46<56>
2780:5C 22 06 46 0C 22 06 46<A5>
2788:34 22 06 46 5C 22 06 46<54>
2790:EC 23 06 46 0F 26 06 46<15>
2798:5C 22 06 46 4C 23 06 46<73>
27A0:74 23 06 46 5C 22 86 42<47>
27A8:2E 48 02 02 41 F9 47 01<73>
27B0:05 05 05 05 05 05 05 05<7E>
27B8:05 05 05 05 05 05 06 05<08>
27C0:05 05 05 05 05 05 06 05<10>
27C8:05 05 05 05 05 06 05 05<1A>
27D0:05 05 05 05 05 05 04 08<1F>
27D8:33 23 2F 32 25 1A 10 10<02>
27E0:10 10 10 10 10 00 08 00<CD>
27E8:15 10 10 38 10 10 08 33<4E>
27F0:35 22 1A 11 00 08 00 2C<01>
27F8:29 36 25 33 1A 13 08 02<5F>
2800:05 05 05 05 05 05 05 05<50>
2808:05 05 05 05 05 05 07 05<5C>
2810:05 05 05 05 05 05 07 05<64>
2818:05 05 05 05 05 07 05 05<70>
2820:05 05 05 05 05 05 03 70<D7>
2828:70 C6 0C 22 46 5C 22 06<2F>
2830:46 0C 22 06 46 84 22 06<D9>
2838:46 5C 22 06 46 6F 25 06<5F>
2840:06 06 46 5C 22 06 46 C6<0F>
2848:48 71 49 31 27 06 06 06<58>
2850:46 5C 22 86 42 2E 48 02<CC>
2858:02 41 A6 48 70 70 C6 0C<FD>
2860:22 46 5C 22 06 46 0C 22<74>
2868:06 46 34 22 06 46 5C 22<D9>
2870:06 46 BF 25 06 46 BF 25<67>
2878:06 46 9C 23 06 46 31 27<D1>
2880:06 06 06 46 5C 22 86 42<32>
2888:2E 48 02 02 41 D7 48 70<8A>
2890:70 C6 0C 22 46 5C 22 06<97>
2898:46 31 27 06 06 06 46 5C<4D>
28A0:22 06 46 4C 23 06 46 74<B0>
28A8:23 06 46 9C 23 06 46 24<FF>
28B0:23 06 46 74 23 06 46 5C<2C>
28B8:22 86 42 2E 48 02 02 41<4D>
28C0:0A 49 70 70 C6 0C 22 46<65>
28C8:5C 22 06 46 4C 23 06 46<24>
28D0:74 23 06 46 9C 23 06 46<0B>
28D8:9C 23 06 46 1F 25 06 46<33>
28E0:5C 22 06 46 9C 23 06 46<C7>
28E8:9C 23 06 46 9C 23 86 42<9A>
28F0:2E 48 02 02 41 3D 49 72<C9>
28F8:49 6D 4A A2 FF 9A 20 85<9D>
2900:2C 20 96 40 4C C3 2A 20<00>
2908:44 2D A9 40 8D 0E D4 A2<9E>
2910:FF 9A AD 59 2A 8D 72 48<37>
2918:38 E9 11 8D 75 30 A2 00<EA>
2920:A9 10 9D 5D 48 E8 E0 07<EE>
2928:D0 F6 A9 13 8D 7C 48 A2<AB>
2930:00 8E 1D D0 8E 2F 02 8A<7D>
2938:9D DD 4A E8 E0 8C D0 F8<25>
2940:A9 10 8D 6B 48 8D 6C 48<59>
2948:A9 78 8D 07 D4 A9 03 8D<43>
2950:0A D0 A9 10 8D 0C D0 A9<98>
2958:01 8D 6F 02 A9 94 8D C3<D0>
2960:02 A9 16 8D C0 02 A9 1C<D4>
2968:8D C1 02 A9 06 8D C2 02<38>
2970:A9 78 8D F4 02 A9 A4 8D<8D>
2978:C6 02 AE 75 30 BD D6 2F<8E>
2980:8D C4 02 BD DD 2F 8D C3<95>
2988:02 BD E4 2F 8D 74 30 A9<BE>
2990:16 8D C5 02 A9 FA 8D C7<DF>
2998:02 20 93 3C 20 1F 3D 20<20>
29A0:FD 38 20 26 36 AE 75 30<49>
29A8:BD C7 2F 8D 71 30 BD CF<60>
29B0:2F 8D 70 30 8D 00 D0 8D<80>
29B8:01 D0 30 06 20 00 37 4C<C0>
29C0:3C 4A 20 E8 36 A9 08 8D<F6>
29C8:7E 30 A9 00 8D 1E D0 8D<AC>
29D0:6B 30 8D 86 30 8D 87 30<EE>
29D8:8D 7F 30 8D 81 30 8D C8<13>
29E0:02 8D 7A 30 85 14 A9 03<59>
29E8:8D 1D D0 A9 CD 8D 00 02<F6>
29F0:A9 38 8D 01 02 A9 C0 6E<06>
29F8:4A DC 4A 8D 0E D4 A5 14<97>
2A00:C9 02 D0 FA A9 01 85 14<3B>
2A08:20 13 39 20 81 38 20 55<DC>
2A10:38 20 5E 37 20 D8 33 20<28>
2A18:F2 37 20 59 3C 20 CE 35<FF>
2A20:20 94 35 AD 7A 30 C9 14<D4>
2A28:90 0E A9 14 8D 7A 30 20<08>
2A30:12 35 20 7E 35 20 34 35<76>
2A38:EE 7A 30 20 D6 34 20 59<42>
2A40:3E AD 87 30 C9 00 D0 09<91>
2A48:20 D3 3A 20 5E 3A 20 3A<00>
2A50:3D 20 E6 35 20 40 3C AD<82>
2A58:87 30 C9 00 F0 06 20 95<02>
2A60:3D 20 98 34 20 4B 3A 4C<6E>
2A68:71 4A 00 78 FB 78 00 00<05>
2A70:00 00 00 00 00 00 00 00<C4>
2A78:00 1F 1F 18 18 18 18 18<63>
2A80:18 1F 1F 00 00 00 18 18<D4>
2A88:18 F8 F8 00 00 00 00 00<E5>
2A90:00 F8 F8 18 18 18 00 00<84>
2A98:00 FF FF 00 00 00 00 00<8C>
2AA0:00 FF FF 18 18 18 18 18<7F>
2AA8:18 FF FF 00 00 00 18 18<70>
2AB0:18 18 18 18 18 18 78 18<C2>
2AB8:18 1C 1C 1C 7C 00 00 66<86>
2AC0:3C FF 3C 66 00 00 00 18<F7>
2AC8:18 7E 18 18 00 00 00 00<0C>
2AD0:00 00 00 18 18 30 00 00<27>
2AD8:00 7E 00 00 00 00 00 00<8B>
2AE0:00 00 00 18 18 00 03 06<82>
2AE8:0C 18 30 60 40 00 7F 63<A8>
2AF0:63 63 63 63 7F 00 38 18<A5>
2AF8:18 18 3E 3E 3E 00 7F 03<DB>
2B00:03 7F 60 60 7F 00 7E 06<A6>
2B08:06 7F 07 07 7F 00 70 70<9F>
2B10:70 70 77 7F 07 00 7F 60<86>
2B18:60 7F 03 03 7F 00 7C 6C<E7>
2B20:60 7F 63 63 7F 00 7F 03<66>
2B28:03 1F 18 18 18 00 3E 36<7F>
2B30:36 7F 77 77 7F 00 7F 63<FB>
2B38:63 7F 07 07 07 00 3C 3C<5C>
2B40:3C 00 3C 3C 3C 00 3C 3C<8C>
2B48:3C 00 3C 3C 18 30 06 0C<79>
2B50:18 30 18 0C 06 00 00 7E<10>
2B58:00 00 7E 00 00 00 60 30<6F>
2B60:18 0C 18 30 60 00 7F 63<30>
2B68:03 1F FC 78 DF 79 1C 00<22>
2B70:1C 00 18 18 18 18 18 18<C2>
2B78:18 18 3F 33 33 7F 73 73<EB>
2B80:73 00 7E 66 66 7F 67 67<93>
2B88:7F 00 7F 67 67 60 63 63<36>
2B90:7F 00 7E 66 66 77 77 77<CD>
2B98:7F 00 7F 60 60 7F 70 70<61>
2BA0:7F 00 7F 60 60 7F 70 70<59>
2BA8:70 00 7F 63 60 6F 67 67<AD>
2BB0:7F 00 73 73 73 7F 73 73<5A>
2BB8:73 00 7F 1C 1C 1C 1C 1C<1F>
2BC0:7F 00 0C 0C 0C 0E 0E 6E<3A>
2BC8:7E 00 66 66 6C 7F 67 67<F1>
2BD0:67 00 30 30 30 70 70 70<66>
2BD8:7E 00 67 7F 7F 77 67 67<4D>
2BE0:67 00 67 77 7F 6F 67 67<33>
2BE8:67 00 7F 63 63 67 67 67<DA>
2BF0:7F 00 7F 63 63 7F 70 70<11>
2BF8:70 00 7F 63 63 67 67 67<16>
2C00:7F 07 7E 66 66 7F 77 77<FA>
2C08:77 00 7F 60 7F 03 73 73<C1>
2C10:7F 00 7F 1C 1C 1C 1C 1C<FE>
2C18:1C 00 67 67 67 67 67 67<C0>
2C20:7F 00 67 67 67 67 6F 3E<EE>
2C28:1C 00 67 67 67 6F 7F 7F<58>
2C30:67 00 73 73 73 3E 67 67<7C>
2C38:67 00 67 67 67 7F 1C 1C<29>
2C40:1C 00 7F 66 6C 18 37 67<8E>
2C48:7F 00 1E 18 18 18 18 18<0E>
2C50:1E 00 00 80 FB 80 00 00<95>
2C58:00 00 00 00 00 00 FF FF<AD>
2C60:FF FF FF FF FF 00 FE FE<B9>
2C68:FE FE FE FE FE 00 F7 F7<B3>
2C70:F7 F7 F7 F7 F7 00 FF FF<05>
2C78:66 66 66 66 66 66 66 66<1B>
2C80:66 66 66 66 66 66 66 66<53>
2C88:66 66 66 FF FF 66 00 00<7B>
2C90:00 00 00 00 0F 1F 00 00<DD>
2C98:00 00 00 00 C0 C0 0F 0B<1D>
2CA0:1F 03 0F 02 02 02 C0 C0<22>
2CA8:C0 C0 C0 E0 E0 00 1F 2D<B0>
2CB0:4F 3D 0F 1D 3F 7D C0 A0<A2>
2CB8:90 E0 80 C0 E0 F0 00 00<0B>
2CC0:00 00 00 00 C0 C0 05 05<2C>
2CC8:05 05 05 05 1D 1D 05 05<41>
2CD0:05 1D 1D 01 01 01 02 07<BC>
2CD8:FD FF 45 11 FF 00 00 00<20>
2CE0:55 FF FF FF 42 C6 00 00<9F>
2CE8:40 F0 FE FF 00 00 C2 77<08>
2CF0:3D 1F 07 01 FF 00 00 00<97>
2CF8:55 FF FF FF 42 63 00 00<2A>
2D00:15 BF F8 E0 00 00 00 00<91>
2D08:0A 1F 7F FF 00 00 00 00<0F>
2D10:54 FF 1F 07 00 00 40 E0<88>
2D18:BE FF A2 88 FF 00 47 EE<40>
2D20:BC F0 80 8A FF 00 00 00<E4>
2D28:18 66 3C 5A 99 9B 24 42<FE>
2D30:81 99 66 3C 14 24 00 00<63>
2D38:00 18 7E FF 7E 18 18 7E<3A>
2D40:FF 7E 18 00 00 00 00 00<BC>
2D48:06 FF 1E 18 24 42 00 00<D0>
2D50:00 00 FC 80 47 81 00 00<62>
2D58:00 00 00 00 00 00 00 00<B2>
2D60:00 00 3C 18 42 7E FF FF<AC>
2D68:7E 3C 42 24 81 3C 3C 3C<C8>
2D70:41 A4 76 C3 DF C3 FB FB<32>
2D78:C3 76 28 10 38 44 44 44<2E>
2D80:44 38 10 92 92 FE 10 10<9A>
2D88:10 7C FF FF 00 FF FF 00<93>
2D90:FF FF 66 24 00 18 7E DB<6F>
2D98:A5 66 24 99 C3 66 18 18<D0>
2DA0:24 66 E0 02 E1 02 00 22<73>
Laenge 11688 Bytes
```

Listing zu »Submission« (Schluß)

Vorsicht Falle!

Sie drehen in einem Autoscooter Ihre Runden. Plötzlich rast Ihnen ein anderer Scooter entgegen. Aber keine Panik, »Crazy Scooter« ist nur ein Spiel.

Jeder fährt gerne einmal mit einem »Bleifuß« auf dem Gaspedal. Aber bitte nicht auf echten Straßen. Wenn Sie sich so richtig austoben möchten, dann starten Sie doch Ihren Computer und laden Sie das Spiel »Crazy Scooter«. Hier können Sie Ihre Reaktion unter Beweis stellen.

»Crazy Scooter« ist zu 100 Prozent in Maschinensprache geschrieben. Es stellt eine Scooterbahn auf dem Rummelplatz dar. Sie steuern das blaue Fahrzeug, der Computer das rosarote. Ihre Aufgabe besteht darin, alle Punkte auf der Fahrbahn einzusammeln. Achten Sie aber auf das computergesteuerte Auto! Bauen Sie einen Crash, ist eines Ihrer drei Leben verloren. Rechts neben der Scooterbahn befindet sich Ihre Spritanzeige. Geht Ihnen der Sprit aus, ist's ebenfalls für ein Auto vorbei.

Sollten Sie dem anderen Auto einmal blitzschnell ausweichen müssen, betätigen Sie einfach den Feuerknopf. Ihr Auto beschleunigt dann augenblicklich, was sich allerdings auf den Spritverbrauch auswirkt. Achten Sie also auf die Benzinzufuhr.

Haben Sie trotz Schwierigkeiten alle Punkte eingesammelt, gelangen Sie in die nächsthöhere Spielstufe. Ab dem dritten Level kommt eine neue Schwierigkeit ins Spiel. In der Spielfeldmitte erscheint eine Zahl, die die Anzahl der einzusammelnden Bonuspunkte angibt. Diese Punkte erscheinen aber ganz willkürlich an irgendeiner Stelle der Bahn. Für jeden eingesammelten Bonuspunkt bekommen Sie 50 Punkte. Sollten Sie eine Spielstufe mit zu wenig Bonuspunkten beenden, verlieren Sie wiederum einen Scooter. Ab Spielstufe sechs wird es noch interessanter. Dann behindern Sie noch Straßensperren, denen Sie geschickt ausweichen müssen.

Jeder Level unterscheidet sich vom vorhergehenden. Immer wieder gibt es neue Hindernisse. Stellen Sie doch eine eigene Highscore-Liste auf, denn nur wenige werden die Stufe zehn erreichen.

(Oliver Schmitt/wb)

PROGRAMM-STECKBRIEF	
Programmname	Crazy-Scooter
Programmtyp	Spiel
Programmiersprache	Maschinensprache
Programmlänge	6030 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk oder Kassettenrecorder und Joystick
Eingabehilfe	AMPEL
Bemerkung	Sehr schnelles Aktion-Spiel
Leserservice	Diskette (SCOOTER.COM)

```
0000:FF FF 40 90 3B 91 20 08<04>
0008:9D 20 EC 91 20 7D 99 20<AB>
0010:1E 92 20 3B 92 20 E9 9C<BA>
0018:A9 EB 8D 30 02 A9 88 8D<AF>
0020:31 02 A9 80 8D F4 02 A0<1E>
0028:50 A2 06 A9 07 20 5C E4<99>
0030:A9 FA 8D 1A 02 A9 C0 8D<15>
0038:0E D4 20 2E 92 20 B0 93<20>
0040:A9 28 8D 01 D2 4C D1 92<39>
0048:AD 84 02 C9 00 D0 0E AD<98>
0050:2B 90 8D 0A 90 A9 3C 8D<44>
0058:00 D2 4C 9C 92 A9 5A 8D<8C>
0060:00 D2 4C 9C 92 A0 02 B1<E4>
0068:CB 4C B8 90 20 45 99 A0<D8>
0070:00 B1 CB AA 20 53 99 4C<5C>
0078:D8 90 A0 28 B1 CB C9 03<28>
0080:F0 4C C9 04 F0 48 C9 09<60>
0088:F0 29 C9 2F F0 28 C9 38<C0>
0090:F0 39 60 20 6F 99 AC 00<DF>
0098:B1 CB AA 20 61 99 E0 03<AB>
00A0:F0 2C E0 04 F0 28 E0 09<E6>
00A8:F0 09 E0 2F F0 08 E0 38<79>
00B0:F0 19 60 4C C0 95 A9 00<63>
00B8:8D 32 90 20 B0 95 EE 32<C1>
00C0:90 AD 32 90 C9 05 D0 F3<20>
00C8:4C 09 92 4C 7F 91 68 68<A3>
00D0:68 68 4C 47 91 AD 05 90<51>
00D8:C9 0B F0 0B C9 07 F0 0C<FC>
00E0:C9 0E F0 0D 4C 30 91 A9<7E>
00E8:02 4C 41 91 A9 01 4C 41<40>
00F0:91 A9 04 4C 41 91 A9 03<49>
00F8:4C 41 91 68 68 AD 09 90<17>
0100:C9 01 3C 91 37 92 F0 09<9C>
0108:4C C8 92 CD 04 90 F0 EF<72>
0110:60 AD 0A D2 29 03 C9 00<F2>
0118:F0 0B C9 01 F0 0C C9 02<41>
0120:F0 0D 4C 6A 91 A9 0B 4C<F6>
0128:6C 91 A9 07 4C 6C 91 A9<0A>
0130:0E 4C 6C 91 A9 0D 8D 05<40>
0138:90 A9 01 8D 09 90 20 0F<2D>
0140:91 A9 00 8D 09 90 4C D1<BD>
0148:92 A9 00 8D 24 90 8D 02<C0>
0150:D2 20 20 9E 20 07 94 20<45>
0158:F1 94 20 74 95 20 EE 9E<D6>
0160:20 39 94 4C B2 9C AD 0F<40>
0168:90 C9 9D F0 01 60 A9 00<EF>
0170:8D 01 D2 8D 03 D2 AD 2C<F4>
0178:90 C9 00 D0 09 20 EE 9E<F4>
0180:20 58 9E 4C 02 93 A9 00<E0>
0188:8D 24 90 20 20 9E 20 71<75>
0190:94 20 EE 9E 20 39 94 A9<44>
0198:01 8D 2D 90 20 31 9D A9<B5>
01A0:00 8D 2D 90 4C 02 93 A9<9E>
01A8:10 8D A9 8A 8D AA 8A 8D<38>
01B0:AB 8A 8D AC 8A 60 A9 00<32>
01B8:8D 00 02 A9 06 8D 01 02<46>
01C0:A9 F9 8D 28 02 A9 9E 8D<E6>
01C8:29 02 A9 01 8D 04 D4 8D<AE>
01D0:23 90 60 AD 2C 90 C9 00<5C>
01D8:D0 01 60 CE 2C 90 AD 2C<14>
01E0:90 18 69 10 8D DD 89 60<29>
01E8:20 4A 96 20 8C 99 20 09<12>
01F0:99 20 F4 9B 20 32 99 60<92>
01F8:A9 FF 8D 1E D0 AD 1F D0<C5>
0200:C9 06 38 92 33 93 D0 F4<16>
0208:60 A9 00 8D 1C 90 8D 1D<92>
0210:90 8D 1E 90 8D 1F 90 A9<15>
0218:01 8D 1A 90 A9 0A 8D 1B<CD>
0220:90 60 A9 14 8D 22 90 AD<7B>
0228:0A 90 C9 04 F0 04 20 68<A4>
0230:92 60 20 A5 92 60 EE 20<AD>
0238:90 AE 20 90 A9 00 9D 00<3D>
0240:83 E0 B4 F0 08 E0 A0 10<42>
0248:01 60 4C 45 94 A9 00 8D<86>
0250:24 90 8D 02 D2 20 20 9E<09>
0258:20 20 94 20 F1 94 20 74<77>
0260:95 20 EE 9E 20 39 94 4C<B5>
0268:B2 9C CE 22 90 AD 22 90<57>
0270:F0 B0 60 EE 20 90 AE 20<E3>
0278:90 A9 00 9D 00 83 E0 B4<FA>
0280:F0 CB EE 20 90 AE 20 90<72>
0288:A9 00 9D 00 83 E0 B4 F0<00>
0290:BC E0 A0 10 B5 60 AD 78<F8>
0298:02 8D 05 90 20 0F 91 20<E8>
02A0:82 90 A9 00 85 4D AD 05<B4>
02A8:90 C9 0B F0 0B C9 07 F0<01>
02B0:0D C9 0E F0 0F 4C FC 92<61>
02B8:20 CD 96 4C D1 92 20 81<25>
02C0:96 4C D1 92 20 72 97 4C<37>
02C8:D1 92 20 1A 97 4C D1 92<54>
02D0:AD 1A 90 C9 09 F0 3A EE<9A>
02D8:1A 90 20 7C 9D 20 1E 92<D5>
02E0:18 AD 1A 90 69 10 8D D5<D1>
02E8:8A CE 1B 90 CE 1B 90 AD<2E>
02F0:1A 90 C9 03 F0 3B C9 04<11>
02F8:F0 3F C9 05 F0 43 C9 06<32>
0300:F0 4C 34 93 2F 94 C9 07<61>
0308:F0 5A C9 08 F0 5E C9 09<C9>
0310:F0 62 4C 77 90 A9 02 8D<5B>
0318:2B 90 8D 1B 90 A9 11 8D<B1>
0320:D5 8A A9 10 8D D6 8A 20<40>
0328:7C 9D 20 1E 92 A9 05 8D<47>
0330:2C 90 4C A7 93 A9 01 8D<2E>
0338:2C 90 4C A7 93 A9 02 8D<28>
0340:2C 90 4C A7 93 A9 02 8D<00>
0348:2B 90 A9 03 8D 2C 90 4C<AF>
0350:A7 93 A9 0A 8D 1B 90 A9<9E>
0358:04 8D 2B 90 A9 01 8D 2C<78>
0360:90 4C A7 93 A9 02 8D 2C<C3>
0368:90 4C A7 93 A9 03 8D 2C<BF>
0370:90 4C A7 93 A9 04 8D 2C<BB>
0378:90 18 69 10 8D DD 89 4C<B9>
0380:77 90 A9 AA 8D 01 D2 A2<E4>
0388:00 BD D2 93 8D 00 D2 A9<1A>
0390:00 85 14 A9 FF 8D 1E D0<3F>
0398:A5 14 C9 02 D0 F5 E8 E0<B3>
03A0:16 D0 E6 60 1D 1F 23 28<3A>
03A8:2D 2F 35 3C 40 48 51 5B<9C>
03B0:60 6C 79 80 90 A2 B6 C1<32>
03B8:D9 F3 A2 00 BD F6 93 9D<BC>
03C0:D5 89 E8 E0 11 D0 F5 60<70>
03C8:27 00 21 00 2D 00 25 04<3C>
03D0:00 04 2F 00 36 00 25 00<B9>
03D8:32 A2 00 BD 15 94 9D DC<4C>
03E0:8A E8 E0 0B D0 F5 60 33<98>
03E8:23 28 37 21 23 28 2B 2F<42>
03F0:30 26 A2 00 BD 2E 94 9D<AF>
03F8:DC 8A E8 E0 0B D0 F5 60<2B>
0400:2F 35 30 94 2B 95 34 00<52>
0408:2F 26 00 26 35 25 2C A9<D1>
0410:00 AA 9D DC 8A E8 E0 0B<AE>
0418:D0 F8 60 A9 EF 8D 03 D2<32>
```

Listing zu »Crazy Scooter«. Bitte mit AMPEL eingeben.

```
0420:AD 25 90 C9 01 F0 10 A9<37>
0428:BA 8D C1 02 A9 01 8D 25<38>
0430:90 A9 00 8D 02 D2 60 A9<F4>
0438:38 8D C1 02 A9 00 8D 25<7B>
0440:90 A9 45 8D 02 D2 60 A2<95>
0448:00 BD 7F 94 9D DD 8A E8<24>
0450:E0 09 D0 F5 60 2E 2F 00<ED>
0458:00 22 2F 2E 35 33 20 12<66>
0460:95 AD 26 90 C9 00 F0 01<B2>
0468:60 20 E0 94 AD 0A D2 29<C7>
0470:01 C9 01 F0 1C 20 D9 94<F1>
0478:BD 35 89 C9 00 F0 01 60<8C>
0480:A9 2F 9D 35 89 8E 27 90<84>
0488:A9 01 8D 26 90 8D 28 90<50>
0490:60 20 D9 94 BD 75 8A C9<BA>
0498:00 F0 01 60 A9 2F 9D 75<68>
04A0:8A 8E 27 90 A9 01 8D 26<C9>
04A8:90 A9 02 8D 28 90 60 AD<2B>
04B0:0A D2 29 0F AA 60 CE 2A<E4>
04B8:90 AD 2A 90 D0 06 A9 64<7E>
04C0:8D 2A 90 60 68 68 60 AD<D4>
04C8:26 90 C9 01 F0 01 60 AD<E2>
04D0:28 90 C9 02 F0 09 A9 00<FE>
04D8:AE 27 90 9D 35 89 60 A9<62>
04E0:00 AE 27 90 9D 75 8A 60<64>
04E8:AD 1A 90 C9 03 10 02 68<30>
04F0:68 60 20 95 95 AD 2F 90<1F>
04F8:C9 00 F0 01 60 20 9F 95<4A>
0500:AD 0A 2C 95 27 96 D2 29<8E>
0508:01 C9 01 F0 1C 20 6D 95<B3>
0510:BD 5F 89 C9 00 F0 01 60<5F>
0518:A9 38 9D 5F 89 8E 30 90<D9>
0520:A9 01 8D 2F 90 8D 31 90<68>
0528:60 20 6D 95 BD 4F 8A C9<66>
0530:00 F0 01 60 A9 38 9D 4F<0C>
0538:8A 8E 30 90 A9 01 8D 2F<53>
0540:90 A9 02 8D 31 90 60 AD<B9>
0548:0A D2 29 0B AA 60 AD 2F<FE>
0550:90 C9 01 F0 01 60 AD 31<83>
0558:90 C9 02 F0 09 A9 00 AE<34>
0560:30 90 9D 5F 89 60 A9 00<E6>
0568:AE 30 90 9D 4F 8A 60 AD<1D>
0570:1A 90 C9 06 10 02 68 68<65>
0578:60 CE 2E 90 AD 2E 90 D0<F2>
0580:06 A9 64 8D 2E 90 60 68<15>
0588:68 60 A9 00 8D 26 90 20<30>
0590:74 95 A9 00 8D 2F 90 4C<D9>
0598:C3 95 EE 0F 90 EE 0B 90<FA>
05A0:AD 0B 90 C9 0A D0 0D A9<B3>
05A8:00 8D 0B 90 A9 10 8D AC<82>
05B0:8A 20 E4 95 18 AD 0B 90<B7>
05B8:69 10 8D AC 8A 60 EE 0C<7D>
05C0:90 AD 0C 90 C9 0A D0 0D<9B>
05C8:A9 00 8D 0C 90 A9 10 8D<6D>
05D0:AB 8A 20 05 96 18 AD 0C<59>
05D8:90 69 10 8D AB 8A 60 EE<CD>
05E0:0D 90 AD 0D 90 C9 0A D0<43>
05E8:0D A9 00 8D 0D 90 A9 10<88>
05F0:8D AA 8A 4C 26 96 18 AD<56>
05F8:0D 90 69 10 8D AA 8A 60<61>
0600:EE 0E 28 96 23 97 90 AD<26>
0608:0E 90 C9 0A D0 0D A9 00<05>
0610:8D 0E 90 A9 10 8D A9 8A<91>
0618:4C 47 96 18 AD 0E 90 69<6E>
0620:10 8D A9 8A 60 4C 47 96<55>
0628:A9 08 8D 00 90 8D 03 90<84>
0630:A9 01 8D 19 90 A9 02 8D<A8>
0638:04 90 8D 16 90 A9 07 8D<37>
0640:05 90 A9 0A 8D 0A 90 A9<05>
0648:FF 8D 13 90 A9 64 8D 2A<B0>
0650:90 8D 2E 90 A9 21 8D 20<08>
0658:90 A9 14 8D 22 90 60 20<D1>
0660:C8 97 AD 01 90 C9 08 D0<5F>
0668:26 20 07 9B 20 9F 90 20<54>
0670:EA 98 A9 00 A8 91 CB A9<2E>
0678:00 8D 02 90 A9 08 8D 03<AC>
0680:90 20 09 99 20 53 99 20<1E>
0688:F0 98 A9 00 8D 01 90 A2<F3>
0690:00 5E 08 80 7E 10 80 E8<50>
0698:E0 08 D0 F5 EE 01 90 20<A2>
06A0:5F 9C A9 02 8D 04 90 20<54>
06A8:9C 91 60 20 14 98 AD 01<52>
06B0:90 C9 00 D0 27 20 26 9B<94>
06B8:20 A6 90 20 EA 98 A9 00<73>
06C0:A0 01 91 CB A9 00 8D 03<FE>
06C8:90 A9 08 8D 02 90 20 09<2C>
06D0:99 20 45 99 20 F0 98 A9<FE>
06D8:08 8D 01 90 A2 00 1E 10<7E>
06E0:80 3E 08 80 E8 E0 08 D0<D6>
06E8:F5 CE 01 90 20 5F 9C A9<6C>
06F0:01 8D 04 90 20 9C 91 60<1A>
06F8:20 66 98 AD 00 90 C9 08<53>
0700:D0 26 24 97 1F 98 20 45<BA>
0708:9B 20 B4 90 20 EA 98 A9<EE>
0710:00 A8 91 CB A9 00 8D 02<49>
0718:90 A9 08 8D 03 90 20 42<3A>
0720:9A 20 61 99 20 FC 98 A9<C2>
0728:00 8D 00 90 A9 00 AE 00<FA>
0730:90 9D 08 80 EE 00 90 AE<A2>
0738:00 90 A0 00 B9 63 9A 9D<A2>
0740:08 80 E8 C8 C0 08 D0 F4<9F>
0748:20 5F 9C A9 04 8D 04 90<3A>
0750:20 9C 91 60 20 A5 98 AD<DD>
0758:00 90 C9 00 D0 27 20 64<22>
0760:9B 20 CD 90 20 EA 98 A9<B7>
0768:00 A0 14 91 CB A9 08 8D<73>
0770:02 90 A9 00 8D 03 90 20<50>
0778:42 9A 20 6F 99 20 FC 98<E6>
0780:A9 08 8D 00 90 CE 00 90<5A>
0788:AE 00 90 A0 00 B9 63 9A<3D>
0790:9D 08 60 E8 C8 C0 08 D0<1F>
0798:F4 A9 00 9D 08 80 20 5F<76>
07A0:9C A9 03 8D 04 90 20 9C<8E>
07A8:91 60 AD 04 90 C9 03 F0<C2>
07B0:26 C9 04 F0 01 60 20 B0<F9>
07B8:99 A9 00 A8 91 CB A9 00<FC>
07C0:8D 02 90 A9 08 8D 03 90<1A>
07C8:20 09 99 20 61 99 20 F0<F6>
07D0:98 A9 00 8D 01 90 60 20<D6>
07D8:B7 99 A9 00 A0 14 91 CB<17>
07E0:A9 00 8D 02 90 A9 08 8D<D1>
07E8:03 90 20 09 99 20 F0 98<97>
07F0:A9 00 8D 01 90 60 AD 04<50>
07F8:90 C9 03 F0 05 C9 04 F0<2F>
0800:23 60 20 98 1B 99 20 BE<6C>
0808:99 A9 00 A0 14 91 CB A9<2E>
0810:08 8D 02 90 A9 00 8D 03<E9>
0818:90 20 09 99 20 45 99 20<8A>
0820:F0 98 A9 08 8D 01 90 60<8D>
0828:20 CC 99 A9 00 A8 91 CB<A7>
0830:A9 08 8D 02 90 A9 00 8D<00>
0838:03 90 20 09 99 20 61 99<29>
0840:20 45 99 20 F0 98 A9 08<FF>
0848:8D 01 90 60 AD 04 90 C9<36>
0850:01 F0 05 C9 02 F0 13 60<5F>
0858:20 19 9A A9 00 A0 01 91<A2>
0860:CB 20 FC 98 A9 00 8D 00<37>
0868:90 60 20 B0 99 A9 00 A8<6A>
0870:91 CB A9 00 8D 02 90 A9<8D>
0878:08 8D 03 90 20 42 9A 20<B6>
0880:53 99 20 FC 98 A9 00 8D<14>
0888:00 90 60 AD 04 90 C9 01<64>
0890:F0 05 C9 02 F0 23 60 20<FF>
0898:20 9A A9 00 A0 01 91 CB<21>
08A0:A9 08 8D 02 90 A9 00 8D<8D>
08A8:03 90 20 42 9A 20 6F 99<E9>
08B0:20 FC 98 A9 08 8D 00 90<63>
08B8:60 20 2E 9A A9 00 A8 91<4D>
08C0:CB 20 6F 99 20 53 99 20<35>
08C8:FC 98 A9 08 8D 00 90 60<7E>
08D0:A9 00 8D 08 90 60 A9 81<13>
08D8:A0 00 91 CB A9 82 C8 91<E7>
08E0:CB 60 A9 81 A0 00 91 CB<86>
08E8:A9 82 A0 14 91 CB 60 A2<3D>
08F0:00 AC 02 90 BD 2A 99 99<BD>
08F8:08 80 C8 E8 E0 08 D0 F4<56>
0900:A9 00 1C 99 17 9A AA AC<16>
0908:03 90 99 08 80 C8 E8 E0<26>
0910:08 D0 F7 60 00 E7 42 FF<1E>
0918:FF 42 E7 00 A9 3C 85 CB<21>
0920:A9 B9 85 CC A9 00 A8 91<7B>
0928:CB E6 CB 20 F0 98 60 38<4B>
0930:A5 CB E9 01 85 CB A5 CC<DD>
0938:E9 00 85 CC 60 18 A5 CB<83>
0940:69 01 85 CB A5 CC 69 00<4B>
0948:85 CC 60 18 A5 CB 69 14<52>
0950:85 CB A5 CC 69 00 85 CC<C8>
0958:60 38 A5 CB E9 14 85 CB<F5>
0960:A5 CC E9 00 85 CC 60 A9<D2>
0968:00 8D 0B 90 8D 0C 90 8D<3A>
0970:0D 90 8D 0E 90 60 A9 00<0F>
0978:8D 01 90 8D 02 90 8D 07<D0>
0980:90 8D 08 90 8D 09 90 8D<76>
0988:0F 90 8D 18 90 8D 25 90<CC>
0990:8D 26 90 8D 2D 90 8D 2F<AD>
0998:90 60 A0 15 B1 CB 4C D0<41>
09A0:99 A0 01 B1 CB 4C D0 99<54>
09A8:20 45 99 A0 00 B1 CB AA<2F>
09B0:20 53 99 4C E8 99 A0 13<93>
09B8:B1 CB C9 03 F0 2F C9 04<F0>
09C0:F0 2B C9 09 F0 21 C9 2F<03>
09C8:F0 20 C9 38 F0 01 60 4C<07>
09D0:7F 91 E0 03 F0 17 E0 04<12>
09D8:F0 13 E0 09 F0 09 E0 2F<C6>
09E0:F0 08 E0 38 F0 E9 60 4C<9A>
09E8:C0 95 4C F0 90 68 68 AD<2C>
09F0:07 90 C9 01 F0 03 4C 08<FE>
09F8:91 68 68 A9 01 8D 08 90<62>
0A00:4C FB 18 9A 13 9B 9A A0<6D>
0A08:14 B1 CB 4C D0 99 20 6F<F8>
0A10:99 A0 00 B1 CB AA 20 61<82>
0A18:99 4C E8 99 20 6F 99 20<5E>
0A20:53 99 A0 00 B1 CB AA 20<44>
0A28:61 99 20 45 99 4C E8 99<CA>
0A30:A2 00 AC 02 90 BD 63 9A<82>
0A38:99 08 80 C8 E8 E0 08 D0<A5>
0A40:F4 A9 00 AA AC 03 90 99<76>
0A48:08 80 C8 E8 E0 08 D0 F7<8D>
0A50:60 5A 7E 5A 18 18 5A 7E<63>
0A58:5A A9 01 8D 07 90 AD 78<88>
0A60:02 C9 0E F0 0C C9 0D F0<F8>
0A68:20 A9 01 8D 08 90 4C FB<7F>
0A70:9A A9 00 8D 06 90 20 82<39>
0A78:90 20 72 97 EE 06 90 AD<C8>
0A80:06 90 C9 10 D0 F0 4C FB<A7>
0A88:9A A9 00 8D 06 90 20 82<21>
0A90:90 20 1A 97 EE 06 90 AD<F5>
0A98:06 90 C9 10 D0 F0 4C FB<81>
0AA0:9A A9 01 8D 07 90 AD 78<20>
0AA8:02 C9 0B F0 0C C9 07 F0<D5>
0AB0:20 A9 01 8D 08 90 4C FB<3E>
0AB8:9A A9 00 8D 06 90 20 82<D1>
0AC0:90 20 CD 96 EE 06 90 AD<0B>
0AC8:06 90 C9 10 D0 F0 4C FB<CF>
0AD0:9A A9 00 8D 06 90 20 82<D9>
0AD8:90 20 81 96 EE 06 90 AD<79>
0AE0:06 90 C9 10 D0 F0 4C FB<C7>
0AE8:9A A9 00 8D 07 90 68 68<BE>
0AF0:68 68 4C D1 92 AD 08 90<92>
0AF8:C9 01 D0 01 60 A9 3C A2<FB>
0B00:89 20 14 9B 0F 9C 83 9B<A8>
0B08:A9 64 20 83 9B A9 8C 20<AA>
0B10:83 9B A9 B4 20 83 9B 60<8F>
0B18:AD 08 90 C9 01 D0 01 60<CE>
0B20:A9 05 A2 8A 20 83 9B A9<95>
0B28:2D 20 83 9B A9 55 20 83<34>
0B30:9B A9 7D 20 83 9B 60 AD<7B>
0B38:08 90 C9 01 D0 01 60 A9<8B>
0B40:CB A2 89 20 90 9B A9 CD<5E>
0B48:20 90 9B A9 CF 20 90 9B<E9>
0B50:A9 D1 20 90 9B 60 AD 08<B5>
0B58:90 C9 01 D0 01 60 A9 D5<35>
0B60:A2 89 20 90 9B A9 D7 20<ED>
0B68:90 9B A9 D9 20 90 9B A9<F9>
0B70:DB 20 90 9B 60 C5 CB D0<95>
0B78:16 E4 CC D0 12 68 68 4C<56>
0B80:6B 9A C5 CB D0 09 E4 CC<76>
0B88:D0 05 68 68 4C B3 9A 60<1D>
0B90:AD 0E 90 CD 1F 90 F0 03<C5>
0B98:10 20 60 AD 0D 90 CD 1E<D5>
0BA0:90 F0 03 10 15 60 AD 0C<ED>
0BA8:90 CD 1D 90 F0 03 10 0A<4C>
0BB0:60 AD 0B 90 CD 1C 90 10<92>
0BB8:01 60 AD 0B 90 8D 1C 90<44>
0BC0:AD 0C 90 8D 1D 90 AD 0D<8A>
0BC8:90 8D 1E 90 AD 0E 90 8D<CE>
0BD0:1F 90 A2 00 A0 04 18 BD<88>
0BD8:1C 90 69 10 99 BC 8A E8<BF>
0BE0:88 E0 04 D0 F1 60 A9 3E<CC>
0BE8:8D 2F 02 A9 01 8D 6F 02<07>
0BF0:A9 A8 8D 11 90 A9 78 8D<28>
0BF8:00 D0 8D 10 90 A9 80 8D<63>
0C00:07 D4 10 9C 0B 9D 20 11<7C>
0C08:9D 20 4A 9C A9 00 AA 9D<27>
0C10:00 84 E8 D0 FA 20 38 9C<5F>
0C18:A9 03 8D 1D D0 60 00 E7<B3>
0C20:42 FF FF 42 E7 00 5A 7E<87>
0C28:5A 18 18 5A 7E 5A A0 00<B0>
0C30:AE 11 90 B9 28 9C 9D 00<EB>
```

Listing zu »Crazy Scooter«. Bitte mit AMPEL eingeben. (Fortsetzung)

```
0C38:84 E8 C8 C0 08 D0 F4 60<C8>
0C40:A9 4C 8D C0 02 A9 BC 8D<16>
0C48:C4 02 A9 BA 8D C1 02 A9<B0>
0C50:78 8D C6 02 60 A9 00 8D<52>
0C58:14 90 EE 14 90 A9 00 8D<09>
0C60:15 90 EE 15 90 AD 15 90<9F>
0C68:D0 F8 4C D3 9C 20 8D 9C<C0>
0C70:AD 14 90 CD 0A 90 D0 E2<53>
0C78:A9 0A 8D 0A 90 20 88 94<E2>
0C80:4C 1C 95 AD 04 D0 C9 04<F9>
0C88:F0 05 C9 05 F0 01 60 A9<30>
0C90:00 8D 01 D2 8D 03 D2 A9<CE>
0C98:01 8D 24 90 20 20 9E 20<AE>
0CA0:F1 94 20 74 95 20 EE 9E<F8>
0CA8:20 31 9D 20 E8 93 20 9E<1A>
0CB0:9B 20 2E 92 20 7C 9D 20<39>
0CB8:DD 91 20 7D 99 20 1E 92<52>
0CC0:20 E9 9C 20 73 9D 4C 77<DA>
0CC8:90 CE 13 90 AD 13 90 C9<11>
0CD0:00 F0 03 4C 77 9C AD 1B<AA>
0CD8:90 8D 13 90 4C 00 7C A9<31>
0CE0:11 8D D5 8A A9 01 8D 1A<34>
0CE8:90 A9 03 8D 21 90 A9 0A<92>
0CF0:8D 1B 90 A9 04 8D 2B 90<4A>
0CF8:A9 00 8D 2C 90 60 A9 77<27>
0D00:85 C9 0C 9D 07 9E A9 9C<06>
0D08:85 CA 60 A9 00 AA 9D 00<BF>
0D10:83 E8 D0 FA A9 0C A2 22<7A>
0D18:9D 00 83 E8 E0 B5 D0 F8<E8>
0D20:A9 04 8D 0C D0 A9 C6 8D<1F>
0D28:05 D0 60 CE 21 90 AD 21<B3>
0D30:90 C9 00 D0 05 68 68 4C<27>
0D38:B5 9C C9 02 F0 0C C9 01<0F>
0D40:F0 10 A9 00 8D CA 8A 4C<5E>
0D48:5D 9D A9 00 8D CC 8A 4C<01>
0D50:5D 9D A9 00 8D CB 8A AD<66>
0D58:2D 90 C9 01 F0 0E AD 0F<F5>
0D60:90 48 20 1E 92 68 8D 0F<96>
0D68:90 4C 77 90 60 A9 0A 8D<E8>
0D70:CB 8A 8D CC 8A 60 A0 35<6B>
0D78:A2 89 A9 11 20 F1 9D A0<EA>
0D80:5F A9 0D 20 F1 9D A0 89<24>
0D88:A9 09 20 F1 9D A0 B3 A9<40>
0D90:05 20 F1 9D A0 03 A2 8A<05>
0D98:A9 05 20 F1 9D A0 29 A9<BE>
0DA0:09 20 F1 9D A0 4F A9 0D<93>
0DA8:20 F1 9D A0 75 A9 11 20<39>
0DB0:F1 9D A0 49 A9 0F 20 05<0A>
0DB8:9E A0 73 A9 0B 20 05 9E<03>
0DC0:A0 9D A9 07 20 05 9E A0<41>
0DC8:C7 A9 03 20 05 9E A0 CB<BB>
0DD0:A9 03 20 05 9E A0 A5 A9<95>
0DD8:07 20 05 9E A0 7F A9 0B<58>
0DE0:20 05 9E A0 59 A9 0F 20<49>
0DE8:05 9E 60 84 CB 86 CC 8D<09>
0DF0:12 90 A0 00 A9 09 91 CB<82>
0DF8:C8 CC 12 90 D0 F8 60 84<47>
0E00:CB 8D 08 9E 03 9F 12 90<16>
0E08:A9 89 85 CC A2 00 A0 00<BC>
0E10:A9 09 91 CB 20 61 99 E8<3F>
0E18:EC 12 90 D0 F3 60 A9 8F<A7>
0E20:8D 01 D2 20 73 9E 20 7E<E7>
0E28:9E 20 89 9E AD 24 90 C9<E4>
0E30:00 F0 0F A9 00 A8 AE 11<C3>
0E38:90 9D 00 84 E8 C8 C0 08<71>
0E40:D0 F7 A9 00 AA 9D 08 80<BB>
0E48:E8 E0 10 D0 F8 20 58 9E<EC>
0E50:A9 00 8D 01 D2 60 A9 00<E5>
0E58:A8 91 CB A0 01 B1 CB C9<2D>
0E60:82 D0 07 A9 00 A0 01 91<C2>
0E68:CB 60 A9 00 A0 14 91 CB<B2>
0E70:60 A9 94 85 CF A9 9E 85<9F>
0E78:D0 4C AC 9E A9 9C 85 CF<C0>
0E80:A9 9E 85 D0 4C AC 9E A9<57>
0E88:A4 85 CF A9 9E 85 D0 4C<7B>
0E90:AC 9E 00 08 20 0A 50 14<86>
0E98:04 00 52 C0 0D B1 84 28<49>
0EA0:23 8C 54 02 20 81 00 42<65>
0EA8:11 48 AD 24 90 C9 00 F0<C6>
0EB0:10 A0 00 AE 11 90 B1 CF<CC>
0EB8:9D 00 84 E8 C8 C0 08 D0<53>
0EC0:F5 A0 00 B1 CF 99 08 80<F7>
0EC8:C8 C0 08 D0 F6 A0 00 B1<D6>
0ED0:CF 99 10 80 C8 C0 08 D0<D0>
0ED8:F6 A2 00 E8 A0 00 AD 0A<37>
0EE0:D2 8D 00 D2 C8 D0 F7 E0<75>
0EE8:50 D0 F0 60 A9 00 85 14<96>
0EF0:A5 14 C9 FA D0 FA 60 CE<B1>
0EF8:23 90 AD 23 90 C9 00 F0<F9>
0F00:06 8D 04 9F 25 9F 04 D4<3E>
0F08:4C 20 9F A9 08 8D 04 D4<01>
0F10:8D 23 90 EE EF 88 AD EF<6E>
0F18:88 C9 8D D0 05 A9 00 8D<B3>
0F20:EF 88 A9 01 8D 1A 02 60<ED>
0F28:00 8B A3 8B 00 00 00 00<46>
0F30:00 A3 B2 A1 BA B9 00 B3<DB>
0F38:A3 AF AF B4 A5 B2 00 00<50>
0F40:00 00 00 24 25 33 29 27<FF>
0F48:2E 25 24 00 22 39 00 2F<30>
0F50:2C 29 36 25 32 00 33 23<7A>
0F58:28 2D 29 24 34 00 00 00<5E>
0F60:23 2F 30 39 32 29 27 28<81>
0F68:34 00 0B 00 11 19 18 15<34>
0F70:00 22 39 00 30 32 2F 30<12>
0F78:21 27 21 2E 24 21 00 33<C8>
0F80:2F 26 34 37 21 32 25 00<B5>
0F88:00 00 00 00 A3 B2 A1 BA<7F>
0F90:B9 00 B3 A3 AF AF B4 A5<DC>
0F98:B2 00 00 00 00 00 25 38<11>
0FA0:23 2C 35 33 29 36 00 26<5D>
0FA8:2F 32 00 28 21 30 30 39<D0>
0FB0:00 23 2F 2D 30 35 34 25<23>
0FB8:32 00 00 00 00 00 A3 B2<E7>
0FC0:A1 BA B9 00 B3 A3 AF AF<E1>
0FC8:B4 A5 B2 00 00 00 00 00<60>
0FD0:EB 88 E6 89 70 30 30 D6<14>
0FD8:00 88 C6 20 89 86 86 86<45>
0FE0:86 86 86 86 86 86 86 86<75>
0FE8:86 86 86 86 86 86 86 86<ED>
0FF0:86 86 06 06 41 EB 88 00<D8>
0FF8:00 00 00 00 00 00 00 00<16>
1000:00 00 00 00 00 00 00 00<20>
1008:00 05 03 03 03 03 03 03<27>
1010:03 03 03 03 03 03 03 03<30>
1018:03 03 03 07 00 04 09 09<76>
1020:09 09 09 09 09 00 00 00<01>
1028:09 09 09 09 09 09 09 04<43>
1030:00 04 09 05 03 03 03 03<EF>
1038:03 00 00 00 03 03 03 03<05>
1040:03 07 09 04 00 04 09 04<2B>
1048:09 09 09 09 09 09 09 09<68>
1050:09 09 09 09 09 04 09 04<57>
1058:00 04 09 04 09 05 03 03<40>
1060:03 00 00 00 03 03 03 07<33>
1068:09 04 09 04 00 04 09 04<95>
1070:09 04 09 09 09 09 09 09<47>
1078:09 09 09 04 09 04 09 04<0F>
1080:00 04 09 04 09 04 09 05<72>
1088:03 00 00 00 03 07 09 04<74>
1090:09 04 09 04 00 04 09 04<BD>
1098:09 04 09 04 09 09 09 09<23>
10A0:09 04 09 04 09 04 09 04<14>
10A8:00 04 09 00 09 00 09 00<45>
10B0:09 05 03 07 09 00 09 00<C1>
10B8:09 00 09 04 00 04 09 00<E0>
10C0:09 00 09 00 09 04 00 04<E2>
10C8:09 00 09 00 09 00 09 04<EC>
10D0:E7 89 E2 8A 00 04 09 00<8D>
10D8:09 00 09 00 09 06 03 08<0D>
10E0:09 00 09 00 09 00 09 04<03>
10E8:00 04 09 04 09 04 09 04<D8>
10F0:09 09 09 09 09 04 09 04<F6>
10F8:09 04 09 04 00 04 09 04<25>
1100:09 04 09 06 03 00 00 00<41>
1108:03 08 09 04 09 04 09 04<7D>
1110:00 04 09 04 09 04 09 09<07>
1118:09 09 09 09 09 09 09 04<F4>
1120:09 04 09 04 00 04 09 04<4F>
1128:09 06 03 03 03 00 00 00<F8>
1130:03 03 03 08 09 04 09 04<E3>
1138:00 04 09 04 09 04 09 09<44>
1140:09 09 09 09 09 09 09 09<62>
1148:09 04 09 04 00 04 09 06<79>
1150:03 03 03 03 03 00 00 00<5D>
1158:03 03 03 03 03 08 09 04<9B>
1160:00 04 09 09 09 09 09 09<BC>
1168:09 09 09 09 09 09 09 09<8A>
1170:09 09 09 04 00 06 03 03<DB>
1178:03 03 03 03 03 03 03 03<9A>
1180:03 03 03 03 03 03 03 08<A7>
1188:00 00 33 23 2F 32 25 00<CF>
1190:00 00 00 00 00 00 10 10<E2>
1198:10 10 10 00 00 00 28 29<42>
11A0:27 28 33 23 2F 32 25 00<85>
11A8:00 00 10 10 10 10 10 00<AE>
11B0:00 00 23 21 32 33 00 0A<B1>
11B8:0A 0A 00 00 2C 25 36 25<E5>
11C0:2C 00 11 00 00 00 00 00<1B>
11C8:00 00 00 00 00 00 00 00<EA>
11D0:E3 8A FF 8A 00 00 00 00<4F>
11D8:00 00 00 00 00 00 00 00<FA>
11E0:00 00 00 00 00 00 00 00<02>
11E8:00 00 00 00 00 00 00 00<0A>
11F0:00 00 06 2B 06 48 8A 48<33>
11F8:A6 D1 BD 14 06 8D 0A D4<E5>
1200:8D 16 D0 E6 D1 68 AA 68<B0>
1208:40 7E 7C 7A 78 76 74 76<0B>
1210:78 7A 7E 7A 78 76 74 76<28>
1218:78 7A 7C 7E B8 BA BC 38<82>
1220:38 00 80 FB 80 00 00 00<24>
1228:00 00 00 00 00 00 E7 42<5C>
1230:FF FF 42 E7 00 00 00 00<4A>
1238:00 00 00 00 00 00 00 00<5C>
1240:FF FF 00 00 00 18 18 18<48>
1248:18 18 18 18 18 00 00 00<A3>
1250:1F 1F 18 18 18 18 18 18<31>
1258:1F 1F 00 00 00 00 00 00<53>
1260:F8 F8 18 18 18 18 18 18<88>
1268:F8 F8 00 00 00 00 00 00<86>
1270:18 18 00 00 00 5A 7E 5A<63>
1278:18 5A 7E 5A 00 3C 42 99<92>
1280:A1 A1 99 42 3C 18 18 18<5F>
1288:00 00 00 00 00 02 06 0C<CC>
1290:18 30 60 40 00 00 18 0C<19>
1298:FE 0C 18 00 00 00 00 00<A1>
12A0:00 00 00 00 00 7C C6 C6<08>
12A8:C6 C6 C6 7C 00 18 38 18<9A>
12B0:18 18 18 7E 00 7C C6 0C<4B>
12B8:18 30 60 FE 00 7C C6 06<44>
12C0:0C 06 C6 7C 00 0C 1C 3C<91>
12C8:6C CC FE 0C 00 FE 80 80<8F>
12D0:FC 06 C6 7C 00 7C C6 C0<22>
12D8:FC C6 C6 7C 00 FE 06 0C<FC>
12E0:18 30 30 30 00 7C C6 C6<48>
12E8:7C C6 C6 7C 00 7C C6 C6<C1>
12F0:7E 06 C6 7C 00 00 00 00<45>
12F8:00 00 00 00 00 00 00 00<1C>
1300:00 00 00 00 00 00 00 00<26>
1308:00 00 00 00 00 00 00 00<2E>
1310:00 00 00 00 00 00 00 00<36>
1318:00 00 00 00 00 00 00 00<3E>
1320:00 FC 80 D7 81 00 00 00<CE>
1328:00 00 00 00 00 00 00 00<4E>
1330:00 FE E2 8E FE 82 82 82<77>
1338:00 FC E6 8E FC 8E E6 FC<D0>
1340:00 FE F0 80 80 80 F0 FE<CB>
1348:00 F8 EC C6 82 C6 EC F8<39>
1350:00 FE F0 80 F0 80 F0 FE<57>
1358:00 FE F0 80 F0 80 80 80<F0>
1360:00 FE F0 80 9E 82 F2 FE<E4>
1368:00 C6 C6 82 FE 82 C6 C6<3C>
1370:00 FC 30 30 30 30 30 FC<35>
1378:00 06 06 06 42 42 66 3C<61>
1380:00 C6 CC 98 F0 98 CC C6<56>
1388:00 C0 C0 C0 80 80 8E FE<AF>
1390:00 82 C6 EE BA 92 C6 C6<18>
1398:00 C6 E6 B6 92 DA CE C6<A7>
13A0:00 FE DE C2 C6 86 F6 FE<60>
13A8:00 FE E2 8E FE 80 80 80<D0>
13B0:00 FE 82 82 82 8E 8E FE<0A>
13B8:00 FE E2 8E FE 98 8C 86<5E>
13C0:00 FE F0 80 FE 02 1E FE<9E>
13C8:00 FE 38 38 38 10 10 10<2B>
13D0:00 82 82 82 C6 C6 C6 FE<7E>
13D8:00 C6 C6 82 C6 6C 38 10<A5>
13E0:00 C6 C6 92 BA EE C6 82<34>
13E8:00 C6 EE 7C 38 7C EE C6<86>
13F0:00 CC CC 48 78 30 30 30<13>
13F8:00 FE C6 0C 38 60 C6 FE<DD>
1400:00 50 06 56 06 A9 00 85<AA>
1408:D1 4C 62 E4 00 7C FB 7C<96>
1410:AD 18 90 C9 01 F0 1E C9<E0>
1418:02 F0 1D C9 03 F0 1C C9<76>
1420:04 F0 1B AD 16 90 C9 01<F9>
1428:F0 29 C9 02 F0 13 C9 04<2D>
1430:F0 33 4C 67 7C 4C CE 7D<24>
1438:4C F0 7D 4C FC 7E 4C 21<05>
1440:7F 20 9B 7C 20 16 7E EE<0B>
1448:10 90 AD 10 90 8D 00 D0<79>
```

Listing zu »Crazy Scooter«. Bitte mit AMPEL eingeben. (Fortsetzung)

```
1450:6C C9 00 20 C2 7C 20 6D<0F>
1458:7D CE 10 90 AD 10 90 8D<0F>
1460:00 D0 6C C9 00 20 E9 7C<85>
1468:20 54 7E 20 79 7C EE 11<05>
1470:90 20 89 7C 6C C9 00 20<78>
1478:10 7D 20 98 7E 20 79 7C<1E>
1480:CE 11 90 20 89 7C 6C C9<9D>
1488:00 AE 11 90 A9 00 A8 9D<6A>
1490:00 84 E8 C8 C0 08 D0 F7<C7>
1498:60 AE 11 90 A0 00 B9 64<4B>
14A0:7F 9D 00 84 E8 C8 C0 08<8F>
14A8:D0 F4 60 A9 03 8D 16 90<46>
14B0:A2 B8 A0 A8 20 37 7D A2<7E>
14B8:A8 A0 98 20 37 7D A2 98<08>
14C0:A0 88 20 37 7D A2 88 A0<25>
14C8:78 20 37 7D A9 02 8D 16<E1>
14D0:90 60 A9 04 8D 16 90 A2<AC>
14D8:38 A0 28 20 37 7D A2 48<47>
14E0:A0 38 20 37 7D A2 58 A0<98>
14E8:48 20 37 7D A2 68 A0 58<24>
14F0:20 37 7D A9 01 8D 16 90<0A>
14F8:60 A9 02 8D 16 90 A2 38<33>
1500:A0 A8 20 37 7D A2 48 A0<7E>
1508:98 20 37 7D FC 7C F7 7D<31>
1510:A2 58 A0 88 20 37 7D A2<36>
1518:68 A0 78 20 37 7D A9 04<4C>
1520:8D 16 90 60 A9 01 8D 16<86>
1528:90 A2 B8 A0 28 20 37 7D<40>
1530:A2 A8 A0 38 20 37 7D A2<07>
1538:98 A0 48 20 37 7D A2 88<88>
1540:A0 58 20 37 7D A9 03 8D<A1>
1548:16 90 60 EC 10 90 D0 20<AE>
1550:CC 11 90 D0 1B 68 68 68<66>
1558:68 AD 16 90 C9 01 F0 0A<E1>
1560:C9 02 F0 06 20 89 7C 4C<B4>
1568:00 7C 20 4A 7F 4C 00 7C<03>
1570:60 AD 19 90 C9 01 F0 01<7D>
1578:60 A9 00 8D 19 90 68 68<45>
1580:60 20 5D 7D A9 78 CD 10<23>
1588:90 D0 0F A9 28 CD 11 90<94>
1590:D0 08 20 44 7F C9 01 F0<F6>
1598:5F 60 A9 78 CD 10 90 D0<8C>
15A0:0F A9 58 CD 11 90 D0 08<E6>
15A8:20 44 7F C9 01 F0 27 60<C6>
15B0:A2 78 A0 38 20 AB 7D A2<59>
15B8:78 A0 48 20 AB 7D 60 EC<7D>
15C0:10 90 D0 11 CC 11 90 D0<75>
15C8:0C 68 68 20 44 7F C9 01<80>
15D0:F0 04 4C E4 7D 60 68 68<5D>
15D8:A9 01 8D 18 90 A9 00 8D<DE>
15E0:17 90 20 79 7C CE 11 90<0A>
15E8:20 89 7C EE 17 90 AD 17<1A>
15F0:90 C9 10 F0 25 6C C9 00<FF>
15F8:68 68 A9 02 8D 18 90 A9<34>
1600:00 8D 17 90 20 79 7C EE<09>
1608:11 90 20 89 F8 7D F3 7E<56>
1610:7C EE 17 90 AD 17 90 C9<83>
1618:10 F0 03 6C C9 00 A9 00<07>
1620:8D 18 90 20 4A 7F A9 01<2B>
1628:8D 19 90 6C C9 00 20 5D<9D>
1630:7D A9 78 CD 10 90 D0 0F<12>
1638:A9 A8 CD 11 90 D0 08 20<59>
1640:44 7F C9 01 F0 94 60 A9<4E>
1648:78 CD 10 90 D0 0F A9 78<F2>
1650:CD 11 90 D0 08 20 44 7F<FE>
1658:C9 01 F0 A0 60 A2 78 A0<BE>
1660:98 20 AB 7D A2 78 A0 88<E0>
1668:20 AB 7D 60 20 5D 7D A9<0D>
1670:38 CD 10 90 D0 0F A9 68<5A>
1678:CD 11 90 D0 08 20 44 7F<16>
1680:C9 01 F0 29 60 A9 68 CD<17>
1688:10 90 D0 0F A9 68 CD 11<3D>
1690:90 D0 08 20 44 7F C9 01<6E>
1698:F0 10 60 A2 48 A0 68 20<52>
16A0:D6 7E A2 58 A0 68 20 D6<7D>
16A8:7E 60 4C 12 7F 4C ED 7E<D3>
16B0:20 5D 7D A9 B8 CD 10 90<F3>
16B8:D0 0F A9 68 CD 11 90 D0<C7>
16C0:08 20 44 7F C9 01 F0 62<ED>
16C8:60 A9 88 CD 10 90 D0 0F<1E>
16D0:A9 68 CD 11 90 D0 08 20<01>
16D8:44 7F C9 01 F0 27 60 A2<8D>
16E0:98 A0 68 20 D6 7E A2 A8<E3>
16E8:A0 68 20 D6 7E 60 EC 10<21>
16F0:90 D0 11 CC 11 90 D0 0C<A9>
16F8:68 68 20 44 7F C9 01 F0<CB>
1700:04 4C 12 7F 60 68 68 A9<51>
1708:03 8D 18 90 F4 7E 6B 7F<C2>
1710:20 4A 7F A9 00 8D 17 90<5B>
1718:EE 10 90 AD 10 90 8D 00<C7>
1720:D0 EE 17 90 AD 17 90 C9<47>
1728:10 F0 28 6C C9 00 68 68<95>
1730:A9 04 8D 18 90 20 4A 7F<D5>
1738:A9 00 8D 17 90 CE 10 90<21>
1740:AD 10 90 8D 00 D0 EE 17<D5>
1748:90 AD 17 90 C9 10 F0 03<AE>
1750:6C C9 00 A9 00 8D 18 90<E7>
1758:A9 01 8D 19 90 6C C9 00<7A>
1760:AD 0A D2 29 01 60 A0 00<7F>
1768:AE 11 90 B9 5C 7F 9D 00<CF>
1770:84 E8 C8 C0 08 D0 F4 60<A3>
1778:00 E7 42 FF FF 42 E7 00<67>
1780:5A 7E 5A 18 18 5A 7E 5A<A9>
1788:E0 02 E1 02 40 90 00 00<A7>
Laenge 6030 Bytes
```

Listing zu »Crazy Scooter«. Bitte mit AMPEL eingeben. (Schluß)

Pacman mal 2

Versuchen Sie sich gegen pfiffige Monster. Schlagen Sie sich durch Gänge und sammeln Sie Punkte, alleine oder zu zweit.

Was macht mehr Spaß als Programmieren? Natürlich ein erholsames Spielchen. Und was macht noch mehr Spaß! Wenn zwei Personen gleichzeitig gegeneinander antreten. Bei »Twomaze« handelt es sich um ein Spiel für zwei Personen. Es ist in Maschinensprache geschrieben, verfügt über hervorragende Grafik und ist besonders schnell.

»Twomaze« ist eine Pacman-Variante. Wie auch beim Original, besteht Ihre Aufgabe darin, möglichst viele Punkte zu fressen. Aber Vorsicht! Sie sind nicht der einzige, der sich in den Labyrinthgängen bewegt. Kleine, gierige Monster trachten Ihnen nach dem Leben. Allerdings gibt es noch Kraftpillen, die Sie besonders dann nicht verschmähen sollten, wenn sich Ihnen einer oder mehrere der Bösewichte nähern. Haben Sie rechtzeitig eine Kraftpille verspeist, verfügen Sie über enorme Kräfte und können Ihre Gegner vernichten. Achten Sie dabei aber auf die Zeituhr am rechten Bildschirmrand! Wenn diese nämlich abgelaufen ist, sind Sie wieder ein normaler Pacman und müssen sich mit äußerster Vorsicht durch die Gänge bewegen.

Für das rasante Spiel ist übrigens mindestens ein Joystick nötig. Über die Tastatur lassen sich die Pacmen nicht steuern.

Wer glaubt, daß der gute alte Pacman schon zum alten Eisen zählt, hat sich getäuscht. Bei unserem Listing »Twomaze« gehen nämlich gleich zwei »Pacmänner« gleichzeitig auf Punktejagd (Mampf!). Schnappen Sie sich einen Freund und zwei Joysticks und schon kann die Labyrinth-Hatz mit einer aufregenden, neuen Spiel-Variante losgehen. Pacman-Spieler aller Länder vereinigt euch und tretet in diesem furiosen Simultan-Programm gegeneinander an! Das ideale Spiel für Turniere und Punktspiel-Betrieb (Gobbel-Gobbel-Schluck!).

Zur Eingabe von »Twomaze« verwenden Sie bitte AMPEL. Beachten Sie auch die Hinweise zu diesem Eingabeprogramm. »Twomaze« läßt sich übrigens nicht von Basic aus eintippen. Dies gilt für alle abgedruckten AMPEL-Listings. Und jetzt noch viel Vergnügen bei der Monsterjagd.

(Frank Ostrowski/wb)

PROGRAMM-STECKBRIEF	
Programmname	Twomaze
Programmtyp	Spiel
Programmiersprache	Maschinensprache
Programmlänge	4219 Byte
für Computer	alle
zusätzliche Hardware	Diskettenlaufwerk oder Kassettenrekorder, ein oder zwei Joysticks
Eingabehilfe	AMPEL
Bemerkung	Pacman-Variante für zwei Personen
Leserservice	Diskette (TWOMAZE.COM)

```
0000:FF FF E0 02 E1 02 06 20<16>
0008:00 20 6E 30 00 21 00 20<75>
0010:06 20 A9 3C 8D 02 D3 A9<AF>
0018:06 85 0C A9 20 85 0D A9<62>
0020:01 85 09 A9 00 8D 44 02<3C>
0028:4C FE 21 70 60 42 00 32<33>
0030:02 20 42 0E 21 02 02 02<59>
0038:02 00 02 00 47 82 20 10<09>
0040:42 E6 20 B0 44 B8 39 04<51>
0048:04 04 04 04 04 04 04 04<48>
0050:04 04 04 04 41 1F 20 70<4A>
0058:60 42 00 32 02 A0 44 28<FA>
0060:38 04 04 04 04 04 04 04<7A>
0068:04 04 04 04 04 04 04 04<68>
0070:04 04 04 04 04 04 04 41<AD>
0078:4B 20 EF EE E5 00 70 6C<D4>
0080:61 79 65 72 00 4D 00 00<44>
0088:E5 E1 F3 F9 00 00 F4 F7<80>
0090:EF 00 70 6C 61 79 65 72<04>
0098:00 4D 00 00 E5 E1 F3 F9<75>
00A0:00 00 EF EE E5 00 70 6C<F1>
00A8:61 79 65 72 00 4D 00 EE<8A>
00B0:EF F2 ED E1 EC 00 F4 F7<96>
00B8:EF 00 70 6C 61 79 65 72<1C>
00C0:00 4D 00 EE EF F2 ED E1<ED>
00C8:EC 00 EF EE E5 00 70 6C<EF>
00D0:61 79 65 72 00 4D 00 00<34>
00D8:E8 E1 F2 E4 00 00 F4 F7<60>
00E0:EF 00 70 6C 61 79 65 72<30>
00E8:00 4D 00 00 E8 E1 F2 E4<C6>
00F0:00 00 00 00 00 00 00 00<F0>
00F8:00 33 65 6C 65 63 74 00<98>
0100:27 61 6D 65 00 0D 00 30<46>
0108:72 65 73 73 00 34 72 69<30>
0110:67 67 65 72 00 00 00 00<43>
0118:00 00 00 00 4C 4C 4C 00<44>
0120:4C 00 00 00 4C 00 4C 4C<8F>
0128:4C 4C 00 4C 4C 4C 4C 4C<9E>
0130:00 4C 4C 4C 4C 00 4C 4C<88>
0138:4C 4C 00 4C 4C 4C 4C 00<62>
0140:00 00 00 00 00 4C 00 00<73>
0148:4C 00 00 00 4C 00 4C 00<6B>
0150:00 4C 00 4C 00 4C 00 4C<A7>
0158:00 4C 00 00 4C 00 00 00<CF>
0160:4C 4C 00 4C 00 00 00 00<DF>
0168:00 00 00 00 00 4C 60 00<9B>
0170:4C 00 4C 00 4C 00 4C 00<7A>
0178:00 4C 06 4C 00 4C 00 4C<7F>
0180:00 4C 4C 4C 4C 00 00 4C<82>
0188:4C 00 00 4C 4C 4C 00 00<05>
0190:00 00 00 00 00 4C 00 00<C3>
0198:4C 00 4C 00 4C 00 4C 00<45>
01A0:00 4C 00 4C 00 00 00 4C<C6>
01A8:00 4C 00 00 4C 00 4C 4C<FC>
01B0:00 00 00 4C 00 00 00 00<77>
01B8:00 00 00 00 00 4C 00 00<EB>
01C0:4C 4C 4C 4C 4C 00 4C 4C<71>
01C8:4C 4C 00 4C 00 00 00 4C<D4>
01D0:00 4C 00 00 4C 00 4C 4C<24>
01D8:4C 4C 00 4C 4C 4C 4C 00<82>
01E0:00 00 00 00 00 00 00 00<E2>
01E8:00 00 00 28 61 70 70 79<8C>
01F0:0D 23 6F 6D 70 75 74 65<51>
01F8:72 00 0F 0F 00 26 2F 00<7D>
0200:00 00 00 00 00 00 00 00<04>
0208:00 00 20 7F 2B A9 80 85<8A>
0210:B7 A9 04 85 B6 20 92 22<56>
0218:20 40 22 20 68 22 20 44<B2>
0220:28 A9 01 85 B5 20 AF 22<80>
0228:20 62 27 20 4F 2D 20 96<80>
0230:2F 20 AF 22 A9 11 85 B2<10>
0238:85 B4 20 20 23 A2 FF 9A<19>
0240:20 53 25 20 3F 27 20 83<21>
0248:27 4C 34 22 A9 00 8D 0E<09>
0250:D4 A0 00 B9 00 E0 99 00<62>
0258:3C B9 00 E1 99 00 3D B9<AE>
0260:10 2C 99 00 3E B9 00 E3<46>
0268:99 00 3F C8 D0 E5 A9 3C<AC>
0270:8D F4 02 60 A9 30 8D 07<22>
0278:D4 A9 11 8D 6F 02 A9 3E<9E>
0280:8D 2F 02 A9 03 8D 1D D0<36>
0288:A9 00 A8 99 00 34 99 00<02>
0290:35 99 00 36 99 00 37 99<8C>
0298:00 33 C8 D0 EE 60 A0 08<A5>
02A0:B9 9E 22 99 C0 02 88 10<9D>
02A8:F7 60 FF 3F 6F 9F 16 3C<40>
02B0:CA DF 00 43 44 45 46 47<8C>
02B8:48 49 4A A9 00 A2 03 95<28>
02C0:A3 CA 10 FB A0 4F B9 C7<B2>
02C8:22 99 00 32 88 10 F7 20<B1>
02D0:1C 2E 60 00 00 00 00 00<7A>
02D8:00 00 00 00 00 00 00 00<DC>
02E0:00 00 00 2C 25 36 25 2C<17>
02E8:00 00 10 00 00 00 00 00<EE>
02F0:00 00 00 00 00 00 00 00<F4>
02F8:00 00 00 00 00 00 00 00<FC>
0300:00 10 00 00 00 00 00 00<0A>
0308:00 28 29 27 28 00 00 00<E8>
0310:00 00 00 00 10 00 00 00<96>
0318:00 00 00 00 00 00 00 00<1E>
0320:00 10 00 27 21 2D 25 00<A4>
0328:2F 36 25 32 20 27 23 20<1E>
0330:68 24 60 A9 00 A2 04 95<2D>
0338:A8 CA 10 FB 85 AE 85 AF<C9>
0340:85 9E 85 9F 20 10 2D A9<FD>
0348:28 85 80 A9 38 85 81 A2<A7>
0350:17 A0 26 A9 80 91 80 88<EA>
0358:10 FB 18 A5 80 69 28 85<ED>
0360:80 A5 81 69 00 85 81 CA<E5>
0368:D0 E7 A9 F3 85 80 A9 39<06>
0370:85 81 A9 40 8D F3 39 A9<35>
0378:00 8D 51 38 8D 75 38 AD<A2>
0380:0A D2 29 03 85 82 85 83<0F>
0388:0A A8 A5 80 79 23 25 85<81>
0390:84 A5 81 79 24 25 85 85<0D>
0398:A0 00 B1 84 C9 80 D0 29<5A>
03A0:A5 82 18 69 01 91 84 69<1D>
03A8:03 0A A8 A5 80 79 23 25<45>
03B0:85 80 A5 81 79 24 25 85<19>
03B8:81 A9 00 A8 91 80 A5 84<79>
03C0:85 80 A5 85 85 81 4C 73<57>
03C8:23 E6 82 A5 82 29 03 85<90>
03D0:82 C5 83 D0 B3 A0 00 B1<FB>
03D8:80 85 82 98 91 80 24 82<0E>
03E0:70 18 C6 82 A5 82 0A A8<7C>
03E8:38 A5 80 F9 23 25 85 80<AA>
03F0:A5 81 F9 24 25 85 81 4C<99>
03F8:73 23 A6 A1 BD 43 25 85<71>
0400:95 AD 0A D2 29 3F C9 25<99>
0408:B0 F7 A8 C8 84 97 AD 0A<93>
0410:D2 29 1F C9 15 B0 F7 AA<BF>
0418:45 97 29 01 F0 E3 98 18<30>
0420:69 D6 85 80 A9 37 69 00<DF>
0428:85 81 18 A5 80 69 28 85<DF>
0430:80 A5 81 69 00 85 81 CA<4B>
0438:D0 F0 A0 52 B1 80 F0 C1<81>
0440:A2 03 BC 33 25 B1 80 F0<61>
0448:11 BC 37 25 B1 80 BC 3B<30>
0450:25 11 80 BC 3F 25 11 80<2B>
0458:F0 A7 CA 10 E5 A9 00 A0<C6>
0460:52 91 80 C6 95 D0 9A 8C<6C>
0468:52 38 8C 79 38 8C 74 38<51>
0470:8C 9D 38 60 A9 FF 85 80<34>
0478:A9 37 85 81 A0 29 B1 80<6E>
0480:F0 2E A2 50 A0 01 B1 80<29>
0488:F0 01 E8 A0 2A B1 80 F0<CE>
0490:02 E8 E8 A0 51 B1 80 F0<DA>
0498:04 E8 E8 E8 E8 A0 28 B1<DD>
04A0:80 F0 08 E8 E8 E8 E8 E8<F8>
04A8:E8 E8 E8 8A A0 29 91 80<76>
04B0:E6 80 D0 02 E6 81 A5 80<A4>
04B8:C9 97 D0 C0 A5 81 C9 3B<C0>
04C0:D0 BA A9 28 85 80 A9 38<62>
04C8:85 81 A2 17 A0 25 B1 80<94>
04D0:D0 14 E6 9E D0 02 E6 9F<5D>
04D8:A5 A2 CD 0A D2 90 03 A9<55>
04E0:41 2C A9 40 91 80 88 D0<92>
04E8:E5 18 A5 80 69 28 85 80<83>
04F0:A5 81 69 00 85 81 CA D0<6E>
04F8:D3 A9 00 85 4D A9 42 8D<8A>
0500:71 3B 8D 83 3B 8D 95 3B<CF>
0508:8D F3 39 18 A5 9E 69 FE<A8>
0510:85 9E A5 9F 69 FF 85 9F<D4>
0518:A9 00 8D 51 38 8D 75 38<B2>
0520:A9 50 8D 52 38 8D 74 38<9C>
0528:8D 79 38 8D 9D 38 60 02<6B>
0530:00 FE FF 50 00 B0 FF 01<5C>
0538:00 FF FF 28 00 D8 FF 2A<EF>
0540:51 53 7A 02 29 2B 79 29<00>
0548:50 54 7B 2B 79 7B A2 21<7F>
0550:1C 18 15 13 12 11 10 0F<46>
0558:0E 0D 0C 0B 0A 09 08 A5<08>
0560:86 A5 86 29 07 C9 04 F0<EC>
0568:08 A5 8B 29 0F C9 08 D0<FF>
0570:0E A6 86 A4 8B 20 2A 27<05>
0578:A2 00 86 95 20 7E 26 A5<C6>
0580:8A 29 07 C9 04 F0 08 A5<1C>
0588:8F 29 0F C9 08 D0 0E A6<54>
0590:8A A4 8F 20 2A 27 A2 04<F3>
0598:86 95 20 7E 26 A5 B6 85<97>
05A0:95 E6 96 A6 95 B5 86 0A<9C>
05A8:15 8B 29 0F D0 03 20 C6<66>
05B0:25 A6 95 B5 A8 D0 16 B4<31>
05B8:90 A5 96 4A B0 07 B5 86<D4>
05C0:79 17 26 95 86 18 B5 8B<73>
05C8:79 1D 26 95 8B C6 95 10<0C>
05D0:D2 60 A6 95 BD 66 2B 85<48>
05D8:97 B4 8B B5 86 AA 20 2A<F4>
05E0:27 A2 FF 86 99 A2 03 BC<3C>
05E8:23 26 B1 80 C5 97 B0 02<8F>
05F0:A9 00 95 9A B0 02 E6 99<75>
05F8:CA 10 EC A6 95 F0 44 E0<16>
0600:04 F0 32 A5 99 30 1B F0<5A>
0608:0A B5 90 49 01 A8 A9 01<E9>
0610:99 9A 00 AD 0A D2 29 03<05>
0618:A8 A8 B9 9A 00 D0 F4 98<F2>
0620:95 90 60 01 FF 00 00 00<CE>
0628:00 00 00 01 FF 00 00 2A<66>
0630:28 51 01 29 29 24 B3 30<BE>
0638:E9 AD 00 D3 4A 4A 4A 4A<67>
0640:4C 40 26 24 B1 30 DB AD<FA>
0648:00 D3 29 0F 85 97 24 B7<9C>
0650:30 D0 A2 01 46 97 B0 07<BF>
0658:A5 9D D0 03 A0 03 CA 46<F2>
0660:97 B0 07 A5 9C D0 03 A0<A1>
0668:02 CA 46 97 B0 07 A5 9B<B0>
0670:D0 03 A0 01 CA 46 97 B0<93>
0678:07 A5 9A D0 03 A0 00 CA<02>
0680:8A F0 02 A0 04 A6 95 98<85>
0688:95 90 A6 95 A0 29 B1 80<EC>
0690:C9 41 F0 39 C9 40 F0 32<2D>
0698:C9 42 F0 01 60 20 BE 26<C4>
06A0:A9 16 85 AE A9 07 85 AF<33>
06A8:A2 00 BD FC 26 85 80 E8<29>
06B0:BD FC 26 85 81 A0 33 A9<B7>
06B8:4A 91 80 E8 E0 2C D0 EA<D3>
06C0:A9 00 A2 04 95 A8 CA 10<9C>
06C8:FB 60 A9 01 2C A9 0A 86<49>
06D0:AD 20 DF 26 A9 00 A0 29<2F>
06D8:91 80 A9 01 85 B8 48 A5<72>
06E0:9E D0 02 C6 9F C6 9E 68<FD>
06E8:60 A9 01 E0 00 F0 02 A2<52>
06F0:02 AC 0B D4 C0 04 90 F9<AD>
06F8:C0 7B B0 F5 18 75 A3 95<D9>
0700:A3 B5 A4 69 00 95 A4 60<17>
0708:1C 38 44 38 6C 38 94 38<DB>
0710:BC 38 E4 38 0C 39 34 39<30>
0718:5C 39 84 39 AC 39 D4 39<E9>
0720:FC 39 24 3A 4C 3A 74 3A<FE>
0728:9C 3A C4 3A EC 3A 14 3B<5A>
0730:3C 3B 64 3B 8C 3B 98 4A<33>
0738:4A A8 8A 4A 4A 18 79 F0<BE>
0740:26 85 80 B9 F1 26 69 00<21>
0748:85 81 60 A5 9E 05 9F D0<4F>
0750:1C 20 05 2B 20 20 23 20<A7>
0758:62 27 A6 AD A9 64 20 DF<7E>
0760:26 A5 A3 05 A4 D0 FA A5<28>
0768:A5 05 A6 D0 F4 60 A6 B6<FA>
0770:A9 00 85 8A BD 79 27 95<B9>
0778:86 BD 7E 27 95 8B CA 10<D3>
0780:F3 8E 1E D0 60 30 30 78<04>
0788:C0 C0 30 D0 D0 D0 30 A5<AF>
0790:A1 0A AA BD 24 28 A8 BD<C1>
0798:25 28 AA 98 D0 01 CA 88<71>
07A0:E0 FF D0 F7 A5 AE 05 AF<60>
07A8:F0 42 A5 96 29 01 D0 3C<66>
07B0:A5 AE 0A AA BD FC 26 85<47>
07B8:80 BD FD 26 85 81 A9 00<75>
07C0:A0 33 91 80 A6 AE A4 AF<6F>
07C8:88 10 03 A0 07 CA 86 AE<13>
07D0:84 AF 8A D0 04 C0 00 F0<A9>
07D8:13 0A AA BD FC 26 85 80<F3>
07E0:BD FD 26 85 81 B9 A7 22<09>
07E8:A0 33 91 80 AD FC 02 C9<D0>
07F0:21 D0 13 A9 FF 8D FC 02<24>
07F8:20 FA 27 A9 FF CD FC 02<29>
0800:F0 F6 8D FC 02 60 AD 2A<09>
0808:32 C5 B2 D0 05 A2 00 20<84>
0810:C1 2F AD 4A 32 C5 B4 D0<5E>
0818:05 A2 02 20 C1 2F A9 08<6F>
0820:A9 01 2C 1F D0 D0 08 2C<34>
0828:1F D0 F0 FB 4C 1C 22 60<20>
0830:D0 07 82 06 94 05 E2 04<1B>
0838:57 04 E8 03 8D 03 41 03<20>
```

Listing zu »Twomaze«. Bitte mit AMPEL eingeben.

```
0840:01 03 CA 02 9A 02 71 02<AA>
0848:4C 02 2B 02 0E 02 F4 01<C5>
0850:A9 07 A2 28 A0 5D 20 5C<3B>
0858:E4 A9 2C 8D 00 02 A9 2B<57>
0860:8D 01 02 A9 C0 8D 0E D4<E8>
0868:60 20 3B 2B 20 DD 28 20<2F>
0870:6F 2A 20 72 28 20 4F 2B<37>
0878:20 EB 2F 4C 62 E4 A5 14<F7>
0880:4A B0 14 A5 A3 05 A4 F0<DE>
0888:31 48 A5 A3 D0 02 C6 A4<99>
0890:C6 A3 68 A0 2D D0 12 A5<E5>
0898:A5 05 A6 F0 1D 48 A5 A5<6A>
08A0:D0 02 C6 A6 C6 A5 68 A0<6D>
08A8:4D B9 00 32 D0 02 A9 10<90>
08B0:18 69 01 C9 1A B0 04 99<08>
08B8:00 32 60 A9 10 99 00 32<F4>
08C0:88 4C 9D 28 60 EB 29 F7<CD>
08C8:29 03 2A 0F 2A 1B 2A 27<9D>
08D0:2A 33 2A 27 2A 3F 2A 4B<23>
08D8:2A 57 2A 63 2A 3F 30 4B<00>
08E0:30 63 30 57 30 33 30 33<BE>
08E8:30 A5 14 4A 4A 29 06 AA<BA>
08F0:A5 90 0A A8 A9 02 24 14<60>
08F8:D0 02 A0 08 B9 D1 28 8D<4B>
0900:79 29 B9 D2 28 8D 7A 29<EE>
0908:A5 94 0A A8 A9 02 24 14<47>
0910:D0 02 A0 08 B9 D1 28 8D<05>
0918:AD 29 B9 D2 28 8D AE 29<04>
0920:BD B9 28 8D 86 29 BD BA<C1>
0928:28 8D 87 29 BD C1 28 8D<D9>
0930:93 29 BD C2 28 8D 94 29<4D>
0938:BD C9 28 8D A0 29 BD CA<97>
0940:28 8D A1 29 A0 0B A9 00<C2>
0948:99 00 34 99 00 35 99 00<BC>
0950:36 99 00 37 99 00 33 88<C8>
0958:10 EE A4 8B 8C 3D 29 8C<4E>
0960:83 29 A4 8C 8C 40 29 8C<7A>
0968:90 29 A4 8D 8C 43 29 8C<29>
0970:9D 29 A4 8E 8C 46 29 8C<54>
0978:AA 29 A4 8F 8C 49 29 8C<07>
0980:B7 29 A0 0B B9 33 30 A6<F4>
0988:A8 F0 03 B9 6B 2B 99 00<08>
0990:34 B9 EB 29 A6 A9 F0 03<91>
0998:B9 6B 2B 99 00 35 B9 1B<32>
09A0:2A A6 AA F0 03 B9 6B 2B<A2>
09A8:99 00 36 B9 3F 2A A6 AB<F2>
09B0:F0 03 B9 6B 2B 99 00 37<3C>
09B8:B9 33 30 A6 AC F0 03 B9<30>
09C0:6B 2B 99 00 33 88 18 10<DD>
09C8:BB A6 86 E8 8E 00 D0 A6<AB>
09D0:87 E8 8E 01 D0 A6 88 E8<12>
09D8:8E 02 D0 A6 89 E8 8E 03<8A>
09E0:D0 A6 8A E8 8E 07 D0 E8<24>
09E8:E8 8E 06 D0 E8 E8 8E 05<17>
09F0:D0 E8 E8 8E 04 D0 60 1C<84>
09F8:3E 2A 6B 7F 7F 55 55 55<49>
0A00:55 14 36 1C 3E 3A 7B 7F<5C>
0A08:7F 55 54 54 54 16 30 1C<66>
0A10:3E 2A 6B 7F 7F 55 55 55<70>
0A18:55 14 36 1C 3E 2E 6F 7F<4C>
0A20:7F 55 15 15 15 34 06 08<8A>
0A28:1C 3E 7F 3E 1C 1C 3E 7F<DA>
0A30:3E 1C 08 00 08 1C 3E 7F<13>
0A38:3E 3E 7F 3E 1C 08 00 00<AF>
0A40:00 08 1C 3E 7F 7F 3E 1C<46>
0A48:08 00 00 1C 3E 2A 2A 3E<4F>
0A50:7F 55 55 55 55 55 55 00<7F>
0A58:1C 3E 2A 2A 3E 7F 55 55<8B>
0A60:55 55 14 00 00 1C 3E 2A<8D>
0A68:2A 3E 7F 55 55 55 41 00<5E>
0A70:1C 3E 2A 2A 3E 7F 55 55<F3>
0A78:55 55 55 20 D4 2F A5 AE<51>
0A80:05 AF D0 29 AD 0C D0 25<C9>
0A88:B0 F0 05 A2 00 20 88 2B<74>
0A90:AD 08 D0 0D 09 D0 0D 0A<F7>
0A98:D0 0D 0A D0 29 0E 25 B0<91>
0AA0:F0 05 A2 04 20 88 2B A9<80>
0AA8:01 8D 1E D0 60 AD 08 D0<BA>
0AB0:0D 09 D0 0D 0A D0 0D 0A<0E>
0AB8:D0 29 0E 0D 0C D0 25 B0<0B>
0AC0:F0 49 48 AD 0C D0 25 B0<6C>
0AC8:F0 0C A5 B0 4A AA BD 77<4A>
0AD0:2B A2 00 20 DF 26 AD 08<D3>
0AD8:D0 0D 09 D0 0D 0A D0 0D<73>
0AE0:0A D0 29 0E 25 B0 F0 0C<CB>
0AE8:A5 B0 4A AA BD 77 2B A2<0B>
0AF0:04 20 DF 26 68 4A 4A 90<D9>
0AF8:04 A9 C8 85 A9 4A 90 04<61>
0B00:A9 C8 85 AA 4A 90 04 A9<49>
0B08:C8 85 AB A9 01 8D 1E D0<A9>
0B10:60 E6 A0 A5 A0 C9 16 90<FA>
0B18:02 A9 0F 85 A1 49 0F 4A<15>
0B20:85 A2 A0 17 20 9D 28 A5<4F>
0B28:A0 0A 0A 0A 0A 09 06 8D<7F>
0B30:C4 02 49 8E 8D C5 02 60<9E>
0B38:48 AD C5 02 8D 17 D0 AD<63>
0B40:C6 02 8D 18 D0 68 40 A9<12>
0B48:0C 8D 17 D0 A9 00 8D 18<DF>
0B50:D0 A5 14 29 0F 49 DF 8D<B5>
0B58:C6 02 60 A5 AE 05 AF D0<DF>
0B60:0B A2 04 B5 A8 F0 02 D6<0F>
0B68:A8 CA 10 F7 60 00 02 04<39>
0B70:08 00 51 50 50 50 51 00<1E>
0B78:00 00 1C 3E 3E 3E 3E 1C<75>
0B80:00 00 00 00 96 96 64 96<F6>
0B88:64 64 32 A9 00 AA 95 80<1A>
0B90:E8 10 FB 60 B4 A8 D0 31<02>
0B98:A9 19 95 A8 BD 79 27 95<25>
0BA0:86 BD 7E 27 95 8B A9 04<95>
0BA8:95 90 8A F0 1D C6 B3 10<F2>
0BB0:15 78 A5 A3 05 A4 F0 06<DA>
0BB8:20 8B 28 4C A5 2B 58 20<ED>
0BC0:83 27 A5 B3 30 24 20 1C<2E>
0BC8:2E 60 C6 B1 10 15 78 A5<FC>
0BD0:A3 05 A4 F0 06 20 77 28<B2>
0BD8:4C C2 2B 58 20 83 27 A5<EA>
0BE0:B1 30 04 20 1C 2E 60 A2<D8>
0BE8:08 2C A2 27 A0 08 B9 17<D9>
0BF0:23 9D 00 32 CA 88 10 F6<98>
0BF8:A5 B1 25 B3 10 C8 20 40<18>
0C00:2F 06 B7 20 00 2C 20 00<FA>
0C08:2C 4C 1C 22 A9 00 48 20<A2>
0C10:53 25 20 83 27 68 38 E9<73>
0C18:01 D0 F3 60 00 00 20 44<8D>
0C20:44 20 00 00 00 00 20 A8<4A>
0C28:A8 20 00 00 00 30 FC FC<D0>
0C30:FC FC 30 00 FF 00 00 00<43>
0C38:00 00 00 00 FF FF 00 00<44>
0C40:00 00 00 00 FF FF FF 00<4A>
0C48:00 00 00 00 FF FF FF FF<51>
0C50:00 00 00 00 FF FF FF FF<59>
0C58:FF 00 00 00 FF FF FF FF<E0>
0C60:FF FF 00 00 FF FF FF FF<A8>
0C68:FF FF FF 00 FF FF FF FF<D0>
0C70:FF FF FF FF 00 18 3C 7E<EC>
0C78:7E 3C 18 00 FF 81 81 81<41>
0C80:81 81 81 FF FF 00 00 00<B1>
0C88:00 00 00 00 FF 00 00 00<98>
0C90:00 00 00 00 FF 00 00 00<A0>
0C98:00 00 00 00 00 00 00 00<B0>
0CA0:00 00 00 00 10 10 10 10<A9>
0CA8:10 00 00 00 00 00 00 15<DD>
0CB0:15 00 00 00 10 10 10 15<45>
0CB8:15 00 00 00 00 00 00 10<6B>
0CC0:10 10 10 10 10 10 10 10<D0>
0CC8:10 10 10 10 00 00 00 15<04>
0CD0:15 10 10 10 10 10 10 15<70>
0CD8:15 10 10 10 00 00 00 50<B2>
0CE0:50 00 00 00 10 10 10 50<D1>
0CE8:50 00 00 00 00 00 00 55<7D>
0CF0:55 00 00 00 10 10 10 55<E8>
0CF8:55 00 00 00 00 00 00 50<0A>
0D00:50 10 10 10 10 10 10 50<7A>
0D08:50 10 10 10 00 00 00 55<A6>
0D10:55 10 10 10 10 10 10 55<11>
0D18:55 10 10 10 A9 00 A8 99<11>
0D20:00 38 99 00 39 99 00 3A<A5>
0D28:99 00 3B C8 D0 F1 60 AC<21>
0D30:0B D4 C0 04 90 F9 C0 7B<77>
0D38:B0 F5 A9 1F 8D 30 02 A9<60>
0D40:20 8D 31 02 60 AC 0B D4<6F>
0D48:C0 04 90 F9 C0 7B B0 F5<21>
0D50:A9 4B 8D 30 02 A9 20 8D<65>
0D58:31 02 60 20 27 23 A0 26<44>
0D60:A9 00 99 90 39 A9 80 99<F3>
0D68:B8 39 88 10 F3 20 68 24<4A>
0D70:20 23 2D A9 80 85 B7 A9<C0>
0D78:04 85 90 85 94 20 D8 2D<32>
0D80:20 62 27 20 53 25 20 83<DC>
0D88:27 AD 00 D3 29 0F C9 0F<C2>
0D90:F0 38 C9 0E F0 17 C9 0D<25>
0D98:F0 1A C9 0B F0 06 C9 07<2B>
0DA0:F0 02 D0 26 A5 B5 49 01<1D>
0DA8:85 B5 4C AC 2D E6 B5 E6<03>
0DB0:B5 4C AC 2D C6 B5 C6 B5<BA>
0DB8:20 D8 2D 20 53 25 20 83<62>
0DC0:27 AD 00 D3 29 0F C9 0F<E6>
0DC8:D0 FI AD 10 D0 D0 B4 AD<93>
0DD0:10 D0 F0 FB 46 B7 20 39<1A>
0DD8:2D 20 62 27 C6 B1 C6 B3<FF>
0DE0:20 1C 2E 60 A5 B5 10 03<78>
0DE8:18 69 06 C9 06 90 02 E9<11>
0DF0:06 85 B5 AA BD 48 2E 85<98>
0DF8:B1 BD 4E 2E 85 B3 BD 5A<69>
0E00:2E 85 B6 BD 6C 2E 8D 31<35>
0E08:20 BD 72 2E 8D 32 20 BD<BD>
0E10:54 2E 85 8A BD 60 2E 85<83>
0E18:A0 BD 66 2E 8D 17 32 18<46>
0E20:69 01 8D DE 22 20 05 2B<67>
0E28:A2 0E A9 00 9D 00 32 9D<93>
0E30:19 32 CA 10 F7 A0 00 A6<9B>
0E38:B1 A9 4B CA 30 06 99 00<99>
0E40:32 C8 D0 F7 A6 B3 A0 27<37>
0E48:A9 4B CA 30 06 88 99 00<07>
0E50:32 D0 F7 60 05 05 04 04<66>
0E58:03 03 FF 05 FF 04 FF 03<EF>
0E60:00 C0 00 C0 00 C0 03 04<75>
0E68:03 04 03 04 FF FF 01 01<9E>
0E70:03 03 10 10 12 12 14 14<E6>
0E78:6E 82 96 AA BE D2 20 20<77>
0E80:20 20 20 20 00 4A 4A 4A<C0>
0E88:00 4A 4A 4A 4A 00 4A 4A<35>
0E90:4A 4A 4A 00 4A 4A 4A 4A<03>
0E98:4A 00 00 00 00 4A 00 00<03>
0EA0:4A 00 4A 00 4A 00 4A 00<10>
0EA8:4A 00 00 00 4A 00 4A 4A<1A>
0EB0:00 4A 4A 4A 4A 00 4A 00<12>
0EB8:4A 00 4A 00 4A 4A 4A 00<33>
0EC0:4A 00 00 4A 00 4A 00 00<4F>
0EC8:4A 00 4A 00 4A 00 4A 00<B9>
0ED0:4A 00 00 00 00 4A 4A 4A<99>
0ED8:00 4A 00 00 4A 00 4A 00<24>
0EE0:4A 00 4A 00 4A 4A 4A 4A<44>
0EE8:4A 4A 4A 4A 00 4A 00 00<B2>
0EF0:00 4A 00 4A 4A 4A 4A 00<43>
0EF8:4A 4A 4A 4A 4A 00 00 4A<15>
0F00:00 4A 00 00 00 4A 00 4A<23>
0F08:00 00 00 00 4A 00 00 4A<BA>
0F10:4A 00 00 4A 00 00 4A 00<8A>
0F18:4A 00 00 4A 4A 4A 00 00<73>
0F20:4A 4A 4A 4A 4A 00 00 4A<70>
0F28:00 00 4A 00 4A 00 00 4A<23>
0F30:00 00 00 00 4A 00 4A 00<2D>
0F38:4A 4A 4A 4A 00 00 00 4A<05>
0F40:00 00 00 4A 4A 4A 4A 00<03>
0F48:4A 00 00 4A A0 13 B9 78<E2>
0F50:2E F0 03 99 F9 38 B9 8C<A1>
0F58:2E F0 03 99 21 39 B9 A0<EE>
0F60:2E F0 03 99 49 39 B9 B4<61>
0F68:2E F0 03 99 71 39 B9 C8<2D>
0F70:2E F0 03 99 99 39 B9 DC<92>
0F78:2E F0 03 99 39 3A B9 F0<A9>
0F80:2E F0 03 99 61 3A B9 04<15>
0F88:2F F0 03 99 89 3A B9 18<DA>
0F90:2F F0 03 99 B1 3A B9 2C<48>
0F98:2F F0 03 99 D9 3A 88 10<02>
0FA0:AD 60 A2 28 20 A1 2F A2<6B>
0FA8:48 20 A1 2F 60 A0 00 BD<3C>
0FB0:00 32 D9 02 23 90 08 D0<CD>
0FB8:07 E8 C8 C0 06 D0 F0 60<F8>
0FC0:BD 00 32 99 02 23 E8 C8<31>
0FC8:C0 06 D0 F4 60 F6 B1 B5<74>
0FD0:B2 18 69 01 C9 1A 90 02<DA>
0FD8:A9 10 95 B2 20 1C 2E 60<3A>
0FE0:A9 00 A4 AB D0 02 09 08<13>
0FE8:A4 AA D0 02 09 04 A4 A9<45>
0FF0:D0 02 09 02 85 B0 60 A6<8D>
0FF8:B8 BD 2F 30 8D 00 D2 BD<57>
1000:31 30 8D 01 D2 CA 30 02<60>
1008:86 B8 A5 A8 05 AC 8D 02<5E>
1010:D2 F0 02 A9 45 8D 03 D2<1C>
1018:A5 A3 18 69 14 8D 04 D2<BD>
1020:A5 A5 18 69 14 8D 06 D2<36>
1028:A5 A3 F0 02 A9 C6 8D 05<E3>
1030:D2 A5 A5 F0 02 A9 C6 8D<E1>
1038:07 D2 60 00 5F 00 2A 1C<C7>
1040:3E 7F 7F 7F 7F 7F 7F 7F<42>
1048:7F 3E 1C 1C 3E 7F 7C 78<D9>
1050:70 70 78 7C 7F 3E 1C 1C<22>
1058:3E 7F 1F 0F 07 07 0F 1F<5D>
1060:7F 3E 1C 00 22 63 63 63<19>
1068:77 77 7F 7F 7F 3E 1C 1C<53>
1070:3E 7F 7F 7F 77 77 63 63<DE>
1078:63 22 00 00 00 00 00 00<D2>
Laenge 4219 Bytes
```

Listing zu »Twomaze« (Schluß)



AMPEL-Version 1.1

Mit AMPEL wird das Eingeben von reinen Atari-Maschinenprogrammen zum Kinderspiel.

Speziell bei Maschinenprogrammen kommt es darauf an, daß jedes Bytes korrekt eingegeben wird. Schon ein einziger falscher Wert führt meist dazu, daß das eingegebene Programm nicht läuft. Dies hat dann oft eine stundenlange Fehlersuche zur Folge. Deshalb haben wir für Sie »AMPEL« (Atari-Maschinen-Programm-Eingabe-Listing) entwickeln lassen. Und damit das Eintippen für Sie möglichst einfach und reibungslos verläuft, wählen wir eine spezielle Darstellungsform für Maschinenprogramme. Beachten Sie bitte, daß Sie zukünftig unbedingt AMPEL-Version 1.1 verwenden müssen, um Atari-Maschinenprogramme einzugeben. Von Basic aus lassen sich solche Listings nicht eingeben. Verwahren Sie also das Programm zu AMPEL sorgfältig und legen Sie eine zusätzliche Kopie an einen sicheren Platz.

Geben Sie zunächst das nachfolgende Listing ein. Verwenden Sie dazu den Atari-Prüfsummer. Bevor Sie AMPEL mit RUN starten, sollten Sie das eingegebene Basic-Programm auf Diskette oder Kassette speichern. Hat sich nämlich ein Fehler im Maschinenspracheteil eingeschlichen, kann Ihr Computer abstürzen. In dem Fall müßte AMPEL erneut eingegeben werden.

Grünes und rotes Licht mit AMPEL

Nachdem Sie das Programm mit RUN gestartet haben, werden zuerst die DATA-Werke eingePOKEt. Dieser Vorgang nimmt einige Sekunden in Anspruch. Danach müssen Sie die Länge des einzugebenden Maschinenprogramms eingeben. Diesen Wert entnehmen Sie bitte am Ende des abgedruckten Maschinenprogramms. Anschließend geben Sie den Programmnamen an, unter dem Ihr Programm gespeichert wird. Auf dem Bildschirm erscheint jetzt »0000:«, die Aufforderung, mit der Eingabe der ersten Zeile des AMPEL-Listings zu beginnen.

Betrachten wir eine AMPEL-Listing-Zeile aus der Nähe:

```
0000:00 02 00 07 06 07 20 BD <F3A>
```

Die ersten vier Zahlen (hier 0000) stellen sozusagen die Zeilennummer dar. Diese Zahlen sowie den darauffolgenden Doppelpunkt brauchen Sie nicht einzugeben. Danach folgen acht zweistellige Hexadezimalzahlen, die Sie dem Listing entnehmen. Die Eingabe erfolgt ohne Betätigung der RETURN-Taste. Der Cursor springt selbständig von einer Position zur nächsten. Leerzeichen sowie die beiden Zeichen »⟨« und »⟩« fügt AMPEL automatisch ein. Ist die Prüfsumme korrekt – sie muß auch von Ihnen eingegeben werden – springt der Cursor in die nächste Zeile. Falls nicht, ertönt ein akustisches Signal und alle acht Werte, inklusive der Prüfsumme, müssen nochmals eingegeben werden.

Außer den Hex-Tasten (0 bis 9 und A bis F), wird nur noch DELETE BACKSPACE zum Korrigieren des letzten Zeichens und CONTROL S abgefragt. **Bitte beachten Sie den Kasten »Achtung Änderung«!**

Zwischenspeichern

Da Sie ein umfangreiches Maschinenprogramm sicher nicht an einem Tag eintippen möchten, können Sie zu jedem Zeitpunkt mit CONTROL S zwischenspeichern. Dazu betätigen Sie einfach CONTROL S und der Code wird unter dem zuvor eingegebenen Programmnamen gespeichert. Weiterhin wird automatisch die jeweils zuletzt gespeicherte Version in »BACKUP.OBJ« umbenannt. Zuvor wird natürlich eine ältere »BACKUP.OBJ«-Datei gelöscht. Danach können Sie den Computer ausschalten oder AMPEL mit RESET verlassen.

Wenn Sie das nächstemal AMPEL starten, muß wieder die Programmlänge und der entsprechende Name, mit dem Sie Ihre letzte Version gespeichert haben, eingegeben werden. Existiert bereits eine Datei mit dem entsprechenden Namen, wird die Datei automatisch geladen. Weiterhin wird noch die Zeilennummer auf dem Bildschirm angezeigt, ab der Sie mit der Eingabe der Hex-Werte fortfahren müssen.

Speichern Sie lieber öfter mit CONTROL S ab, um auch gegen eventuelle Stromausfälle gewappnet zu sein. Weiterhin empfiehlt es sich aus Sicherheitsgründen, zwei Disketten für Daten vorzusehen. Wenn Sie das komplette Programm eingegeben haben, speichert der Computer nach Eingabe des letzten Bytes das Programm auf Diskette ab und meldet sich danach mit der READY-Meldung.

Wenn Sie Ihr Maschinenprogramm anschließend laden möchten, erfolgt dies vom DOS-Menü aus. Wählen Sie hier die L-Funktion und geben Sie anschließend den entsprechenden Programmnamen ein.

Besondere Hinweise für Kassettenrecorder-Besitzer

Damit AMPEL auch einwandfrei mit einem Kassettenrecorder funktioniert, müssen folgende Programmzeilen geändert werden (beachten Sie bitte auch die Änderungshinweise »Achtung Änderung«):

```
290 F$="C:"   :AD=0
300 ? " Altes File laden J/N "  ;
310 INPUT FR$
320 IF FR$< > "J"  THEN 510
420 TRAP 510:OPEN #1,4,128,F$
590 TRAP 660:CLOSE#1:OPEN #1,8,128,F$
```

Die Zeilen 330 bis 360 entfallen, ebenso die Zeilen 450, 600 und 610

Speichern Sie anschließend die geänderte AMPEL-Version mit CSAVE ab. Somit haben Sie das benötigte Eingabepro-

PROGRAMM-STECKBRIEF	
Programmname	AMPEL
Programmtyp	Eingabehilfe
Programmiersprache	Atari-Basic
Programmlänge	3965 Byte
für Computer	alle
zusätzliche Hardware	Diskettenlaufwerk oder Kassettenrecorder
Eingabehilfe	Prüfsummer
Bemerkung	komfortable Eingabehilfe für reine Maschinenprogramme. Besitzer eines Kassettenrecorders müssen ein zusätzliches Programm eingeben
Leserservice	Diskette (AMPEL.BAS)

gramm bereits vorliegen. Damit aber Binary-DOS-Files (Maschinenprogramme) von Kassette aus geladen und gestartet werden können, müssen Sie ein zusätzliches Hilfsprogramm eingeben. Dazu muß unbedingt AMPEL verwendet werden.

Beginnen Sie also, indem Sie RUN eingeben. Als Programmnamen geben Sie »C:« (bitte den Doppelpunkt nicht vergessen) und auf die Frage nach der Programmlänge 198 ein (die Programmlänge finden Sie stets am Ende eines AMPEL-Listings). Geben Sie Wert für Wert ein, bis sich das »Kassetten-Hilfsprogramm« vollständig im Speicher befindet. Der Computer meldet sich nach dem letzten Byte mit einem Brummton, der Sie auffordert, die RECORD- und PLAY-Tasten des Recorders zu betätigen. Nach Drücken der RETURN-Taste wird das kurze Hilfsprogramm auf Kassette gespeichert. Speichern Sie es von vornherein am Anfang von verschiedenen Kassetten ab, da es Grundvoraussetzung für das Laden von Maschinenprogrammen von Kassette ist. Notieren Sie sich unbedingt den Zählerstand, um es nicht versehentlich zu löschen.



So sieht der Bildschirm bei der Eingabe von Maschinenprogrammen aus. Zwischenspeichern kann man jederzeit mit CONTROL S.

Sicherheit über alles

Mit CONTROL S kann zu jedem Zeitpunkt gespeichert werden. Daraufhin erklingt ein doppelter Signalton. Betätigen Sie dann noch die RECORD- und PLAY-Tasten des Recorders und anschließend die RETURN-Taste. Der Speichervorgang beginnt. Wenn Sie dann mit einem anderen Programm arbeiten möchten, gelangen Sie mit RESET zurück ins Basic.

Wenn Sie zu einem späteren Zeitpunkt mit der Arbeit fortfahren möchten, laden Sie wieder zuerst AMPEL, starten es mit RUN, geben die Programmlänge ein und beantworten die Frage »Altes File laden J/N« mit J. Daraufhin meldet sich der Computer wieder mit einem Brummton. Spulen Sie das Band an den Anfang zurück, betätigen Sie dann noch die PLAY-Taste und RETURN. Jetzt wird der zuletzt eingegebene Programmteil geladen. Anschließend meldet sich der Atari-Computer wieder mit der Zeilennummer, ab der Sie mit der Eingabe fortfahren müssen.

Um später das eingegebene Programm zu starten, muß stets zuerst das »Kassetten-Hilfsprogramm« geladen werden. Nach dem Ladevorgang meldet sich der Computer mit dem Brummton. Anschließend legen Sie bitte Ihre Kassette mit Ihrem Maschinenprogramm in den Recorder ein, drücken die PLAY-Taste des Recorders und während Sie die START- und OPTION-Tasten des Computers drücken, schalten Sie die Stromversorgung ein. Nach dem Brummton nur noch RETURN drücken und Ihr Programm wird geladen und anschließend automatisch gestartet.

Programme schnell geladen

Speziell bei Kassettenrecordern kann es Probleme beim Laden von Maschinenprogrammen geben. Deshalb sollten Sie stets mehrere Sicherheitskopien von Ihren Programmen machen. Hinzu kommt, daß das Laden von Kassette sehr viel Zeit beanspruchen kann, da die Übertragungsgeschwindigkeit sehr langsam ist. Besitzer einer Diskettenstation haben es hier viel besser. Also nicht vergessen: Wenn Sie als Besitzer eines Kassettenrecorders Maschinensprache-Programme laden wollen, müssen Sie stets zuerst das Kassetten-Hilfsprogramm laden. Sollten Sie mit AMPEL auf Schwierigkeiten stoßen, wenden Sie sich an die Redaktion.

(Frank Ostrowski/wb)

Achtung Änderung!

Wenn Sie AMPEL aus Happy-Computer, Ausgabe 12/85 bereits eingetippt haben, müssen Sie eine kleine Korrektur vornehmen, damit die DELETE/BACKSPACE-Funktion korrekt ausgeführt wird. In den Zeilen 1230, 1660, 1750 hat sich jeweils ein kleiner Fehler eingeschlichen. So sollten die Zeilen aussehen:

```
1230 IF A=8135 THEN LIST 1650,1750
1660 DATA 102, 221, 102, 222, 102, 223
1750 DATA 8135
```

Speichern Sie die neue Version auf Ihrer Masterdiskette. Zur Eingabe von reinen Maschinenprogrammen sollten Sie die AMPEL-Version 1.1 verwenden.

```
0000:00 02 00 07 06 07 20 BD<3A>
0008:07 A2 10 A9 03 9D 42 03<D6>
0010:A9 C3 9D 44 03 A9 07 9D<D3>
0018:45 03 A9 04 9D 4A 03 A9<95>
0020:80 9D 4B 03 20 56 E4 30<73>
0028:6D A9 97 A0 07 8D E0 02<22>
0030:8C E1 02 20 9B 07 C9 FF<6A>
0038:D0 5C C8 D0 59 A9 97 A0<75>
0040:07 8D E2 02 8C E3 02 20<97>
0048:9B 07 C9 FF D0 04 C0 FF<75>
0050:F0 EB 9D 44 03 98 9D 45<EF>
0058:03 20 9B 07 38 FD 44 03<05>
0060:9D 48 03 98 FD 45 03 9D<0A>
0068:49 03 FE 48 03 D0 03 FE<DC>
0070:49 03 20 56 E4 30 1F 20<FC>
0078:98 07 AD 53 03 C9 03 D0<F2>
0080:BC A2 10 A9 0C 9D 42 03<32>
0088:20 56 E4 30 09 20 BD 07<F8>
0090:18 6C E0 02 68 68 38 60<85>
0098:6C E2 02 A9 07 A2 10 9D<8D>
00A0:42 03 A9 00 9D 48 03 9D<60>
00A8:49 03 20 56 E4 30 D2 48<42>
00B0:A9 00 9D 48 03 20 56 E4<64>
00B8:30 DA A8 68 60 A9 3C 8D<7D>
00C0:02 D3 60 43 3A 9B 00 00<0F>
Laenge 198 Bytes
```

Listing »Kassetten-Hilfsprogramm« (Bitte mit AMPEL eingeben)

```
100 ? ,"▲**********************"                   <HI>
110 ? ,"▲*▲▲▲▲▲▲Atari-AMPEL▲▲▲*"                   <XO>
115 ? ,"▲*▲(c)▲Happy-Computer▲*"                   <JT>
117 ? ,"▲*▲▲▲▲▲▲Version▲1.1▲▲▲*"                   <OE>
120 ? ,"▲**********************"                   <HM>
130 ?                                              <JU>
140 ? ,"CTRL▲S▲=▲Sichern"                          <ZY>
150 ?                                              <JY>
160 ? "Einen▲Moment▲Geduld▲bitte.";                <CL>
190 POKE 16,64:POKE 53774,64                       <OQ>
230 DIM F$(15),FB$(15),FR$(27)                     <WK>
240 DIM ML$(261),CIO$(83)                          <MH>
250 GOSUB 1000                                     <QT>
260 ? CHR$(156)                                    <DG>
270 ? "Programmlaenge▲";:INPUT L                   <EH>
280 DIM A$(L+8)                                    <GW>
290 ? "Programmname▲";:INPUT F$                    <IR>
300 IF F$="" THEN 290                              <MA>
310 F=0:FOR I=1 TO LEN(F$):F=F+(F$(I,I)=
":"):NEXT I                                        <IR>
320 IF F=0 THEN FR$="D:":FR$(3)=F$:F$=FR
$                                                  <UE>
330 ? "Filename▲";F$                               <TA>
340 FB$="D:BACKUP.OBJ"                             <NW>
350 FR$=F$:FR$(LEN(FR$)+1)=","                     <XT>
360 FR$(LEN(FR$)+1)=FB$(3)                         <IO>
420 AD=0:TRAP 510:OPEN #1,4,0,F$                   <QM>
430 E=USR(ADR(CIO$),1,7,ADR(A$),L)                 <ZH>
440 IF E=1 THEN ? "Schon▲fertig":END               <IN>
450 IF E<>136 THEN ? "Disk-Read-Error▲";
E:END                                              <QL>
460 AD=PEEK(856)+256*PEEK(857)                     <QT>
470 AD=8*INT(AD/8):TRAP 65535                      <VQ>
510 CD=USR(ADR(ML$),ADR(A$),AD)                    <XM>
520 ON CD GOSUB 590,690,710                        <PW>
530 AD=AD+8*(CD=0)                                 <SV>
540 IF AD<=L THEN 510                              <WK>
550 AD=L:GOSUB 590                                 <NI>
560 POKE 16,192:POKE $D20E,192:END                 <DZ>
590 TRAP 600:CLOSE #1:XIO 33,#1,0,0,FB$            <MD>
600 TRAP 610:CLOSE #1:XIO 32,#1,0,0,FR$            <IF>
610 TRAP 660:CLOSE #1:OPEN #1,8,0,F$               <MU>
620 E=USR(ADR(CIO$),1,11,ADR(A$),AD)               <LC>
630 CLOSE #1                                       <LI>
640 IF E<>1 THEN ? "Disk-Error▲";E                 <ZN>
650 TRAP 65535:RETURN                              <FB>
660 CLOSE #1                                       <LO>
670 ? "Disk-Error▲";PEEK(195):RETURN               <XX>
690 ? CHR$(253):? "Pruefsummenfehler"              <PO>
700 RETURN                                         <MR>
710 ? "Fehler!!!!";:LIST 510:END                   <CW>
1000 Q=0                                           <WO>
1010 FOR I=1536 TO 1592                            <NU>
1020 READ A:Q=Q+A:POKE I,A                         <KX>
1030 NEXT I                                        <EY>
1040 IF Q<>4196 THEN ? "Data▲Error":LIST
 1300,1390:END                                     <JV>
1050 Q=0                                           <XD>
1060 FOR I=1 TO 261                                <RZ>
1070 READ A                                        <QI>
1080 IF A>255 THEN GOSUB 1190                      <EL>
1090 ML$(I)=CHR$(A):Q=Q+A                          <MU>
1100 NEXT I                                        <ER>
1110 READ A:IF Q<>A THEN 1200                      <FZ>
1120 Q=0                                           <WW>
1130 FOR I=1 TO 83                                 <JT>
1140 READ A:Q=Q+A:CIO$(I)=CHR$(A)                  <QL>
1150 NEXT I                                        <FG>
1160 IF Q<>10894 THEN ? "Data▲Error▲";:L
IST 1950,2080:END                                  <MP>
1170 RETURN                                        <PQ>
1190 IF Q=A THEN READ A:Q=0:RETURN                 <VA>
1200 ? "Data▲Error"                                <GO>
1210 IF A=7931 THEN LIST 1430,1530                 <IU>
1220 IF A=6585 THEN LIST 1540,1640                 <PX>
1230 IF A=8135 THEN LIST 1650,1750                 <RD>
1240 IF A=7184 THEN LIST 1760,1860                 <XK>
1250 IF A=2222 THEN LIST 1870,1910                 <OU>
1300 DATA▲173,37,228,72,173,36                     <YX>
1310 DATA▲228,72,96,72,74,74                       <WJ>
1320 DATA▲74,74,32,20,6,104                        <ZA>
1330 DATA▲41,15,9,48,201,58                        <JO>
1340 DATA▲144,2,105,6,168,173                      <OH>
1350 DATA▲71,3,72,173,70,3                         <DW>
1360 DATA▲72,152,96,50,31,30                       <CG>
1370 DATA▲26,24,29,27,51,53                        <IL>
1380 DATA▲48,63,21,18,58,42                        <KV>
1390 DATA▲56,190,52                                <PY>
1430 DATA▲169,0,133,213,104,201                    <HX>
1440 DATA▲2,240,16,170,240,5                       <YR>
1450 DATA▲104,104,202,208,251,169                  <RJ>
1460 DATA▲3,44,169,1,133,212                       <BU>
1470 DATA▲96,104,133,215,104,133                   <YK>
1480 DATA▲214,104,133,217,104,133                  <GO>
1490 DATA▲216,24,101,214,133,214                   <UF>
1500 DATA▲165,217,101,215,133,215                  <IU>
1510 DATA▲169,155,32,28,6,165                      <SR>
1520 DATA▲217,32,9,6,165,216                       <LG>
1530 DATA▲7931                                     <MS>
1540 DATA▲32,9,6,169,58,32                         <OX>
1550 DATA▲28,6,169,0,133,218                       <JQ>
1560 DATA▲173,252,2,41,191,162                     <NS>
1570 DATA▲17,221,39,6,240,5                        <DP>
1580 DATA▲202,16,248,48,239,134                    <GQ>
1590 DATA▲219,141,252,2,32,0                       <UW>
1600 DATA▲6,41,127,32,28,6                         <GB>
1610 DATA▲165,219,201,16,240,168                   <IB>
1620 DATA▲144,40,165,218,240,205                   <EF>
1630 DATA▲74,176,5,169,126,32                      <RS>
1640 DATA▲6585                                     <PH>
1650 DATA▲28,6,160,4,70,220                        <YY>
1660 DATA▲102,221,102,222,102,223                  <VE>
1670 DATA▲102,224,102,225,102,226                  <DY>
1680 DATA▲102,227,102,228,136,208                  <QR>
1690 DATA▲235,198,218,184,80,178                   <BA>
1700 DATA▲10,10,10,10,160,4                        <LO>
1710 DATA▲10,38,228,38,227,38                      <UU>
1720 DATA▲226,38,225,38,224,38                     <YP>
1730 DATA▲223,38,222,38,221,38                     <RU>
1740 DATA▲220,136,208,234,230,218                  <OG>
1750 DATA▲8135                                     <LZ>
1760 DATA▲165,218,201,18,240,19                    <WO>
1770 DATA▲201,16,208,4,169,60                      <MK>
1780 DATA▲208,5,74,176,206,169                     <EG>
1790 DATA▲32,32,28,6,184,80                        <JZ>
1800 DATA▲198,165,217,10,101,216                   <ZP>
1810 DATA▲10,101,220,10,101,221                    <UO>
1820 DATA▲10,101,222,10,101,223                    <YP>
1830 DATA▲10,101,224,10,101,225                    <CQ>
1840 DATA▲10,101,226,10,101,227                    <GR>
1850 DATA▲197,228,208,18,169,62                    <NJ>
1860 DATA▲7184                                     <NS>
1870 DATA▲32,28,6,160,7,185                        <LL>
1880 DATA▲220,0,145,214,136,16                     <KJ>
1890 DATA▲248,169,0,44,169,2                       <NU>
1900 DATA▲133,212,96                               <CA>
1910 DATA▲2222                                     <HM>
1950 DATA▲104,201,4,208,66,104                     <KE>
1960 DATA▲104,170,104,104,133,212                  <AX>
1970 DATA▲104,133,214,104,133,213                  <DH>
1980 DATA▲104,133,216,104,133,215                  <HO>
1990 DATA▲160,134,138,201,8,176                    <AX>
2000 DATA▲33,10,10,10,10,170                       <LS>
2010 DATA▲165,212,157,66,3,165                     <VR>
2020 DATA▲213,157,68,3,165,214                     <TO>
2030 DATA▲157,69,3,165,215,157                     <AD>
2040 DATA▲72,3,165,216,157,73                      <SE>
2050 DATA▲3,32,86,228,132,212                      <HM>
2060 DATA▲169,0,133,213,96,160                     <QU>
2070 DATA▲10,170,240,244,104,104                   <SB>
2080 DATA▲202,208,251,240,237                      <JQ>
```

Listing zu »AMPEL«

Prüfsummer paßt auf

Unser Prüfsummer für alle Atari-Computer ist besonders komfortabel. Mit Hilfe dieses Programms lassen sich Listings auf Anhieb korrekt eingeben.

Mag der Atari-Prüfsummer auf den ersten Blick etwas umfangreich erscheinen, der Aufwand lohnt sich. So legt der Prüfsummer zum Beispiel eine zusätzliche Statuszeile am oberen Bildschirmrand an. Diese Zeile ist von Basic aus nicht zu erreichen, also auch nicht zu löschen. Weiterhin können Sie den Prüfsummer auch für die Eingabe von Turbo-Basic XL-Programmen verwenden. Beachten Sie hierzu die besonderen Hinweise!

In dem abgedruckten Listing finden Sie gewisse Zeichen vor, die der Atari nicht kennt. Es handelt sich dabei einerseits um Dreiecke und andererseits um geschweifte Klammern. Die Dreiecke stellen grundsätzlich Leerstellen dar; Texte, die zwischen geschweiften Klammern stehen, respräsentieren Atari-spezifische Grafik- und Sonderzeichen. Ein Beispiel: »ESC CTL=« entspricht dem ASCII-Zeichen 29. Um dieses Zeichen einzugeben, betätigen Sie zunächst die ESC-Taste und dann gleichzeitig die CONTROL- und die »=«-Taste. Daraufhin erscheint das gewünschte Zeichen auf dem Bildschirm. Eine komplette Aufstellung aller Grafik- und Sonderzeichen finden Sie in der Tabelle.

Weiterhin gibt es unterstrichene Zeichen, die eine inverse Darstellung bedeuten. Bevor Sie ein solches Zeichen eingeben, müssen Sie unbedingt die Atari-Taste betätigen. Bild 1 zeigt eine Programmzeile in der üblichen Darstellungsweise und zum Vergleich dazu eine in konvertierter Form.

Geben Sie also zunächst das Prüfsummenprogramm von Basic – oder Turbo-Basic XL – aus ein. Zwar werden sämtliche DATA-Werte nach dem Start mit RUN überprüft, aber »nobody is perfect«.

Kontrollieren Sie also das Prüfsummenprogramm vor dem Start noch einmal. Speichern Sie es dann auf einem Datenträger. Jetzt können Sie das Programm beruhigt mit RUN starten. Aus den DATA-Werten wird nun ein Maschinen-Programm erzeugt. Sollte sich beim Eintippen des Programms ein falscher DATA-Wert eingeschlichen haben, so wird die entsprechende Zeilennummer auf dem Bildschirm angezeigt.

Nach der Korrektur starten Sie das Programm erneut mit RUN. Wurde das Programm fehlerlos eingegeben, stehen folgende Funktionen zur Auswahl:

1. Das Prüfsummen-Programm als Maschinensprach-File auf Diskette schreiben. Nach Betätigung der Taste 1 fragt das Programm nach dem Namen für das Prüfsummenprogramm. Tippen Sie also beispielsweise »PRUEFER.OBJ« ein, so wird nach Druck auf die RETURN-Taste das Prüfsummenprogramm auf Diskette gespeichert. Der Computer schließt den Vorgang mit der READY-Meldung ab.

Um das Prüfcode-Programm verwenden zu können, löschen Sie den Programmspeicher zunächst mit NEW. Vom DOS-Menü aus (einfach DOS eintippen und die RETURN-Taste betätigen) wählen Sie die Option L und geben den Namen des vorher erzeugten Maschinenprogramms ein. Jetzt wird das Prüfsummenprogramm in den RAM-Speicher geladen. Mit B gelangen Sie wieder zurück ins Basic.

Aktiviert wird jetzt das Prüfsummenprogramm mit »?USR(1536)«. Am oberen Bildschirmrand erhalten Sie daraufhin eine zusätzliche Zeile mit dem Schriftzug

```
4850 POSITION 11,2:? "└───────────────────┘":
POSITION 4,5:? "FILENAME     SIZE":POSITI
ON 2,7

4850 POSITION 11,2:? "{CTL Z}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{SHIFT =}":POSITION 4,5:? "FILENA
ME▲▲▲▲SIZE":POSITION 2,7
```

Bild 1. Oben eine Programmzeile, wie sie auf dem Bildschirm erscheint. Darunter dieselbe Zeile in der konvertierten Form.

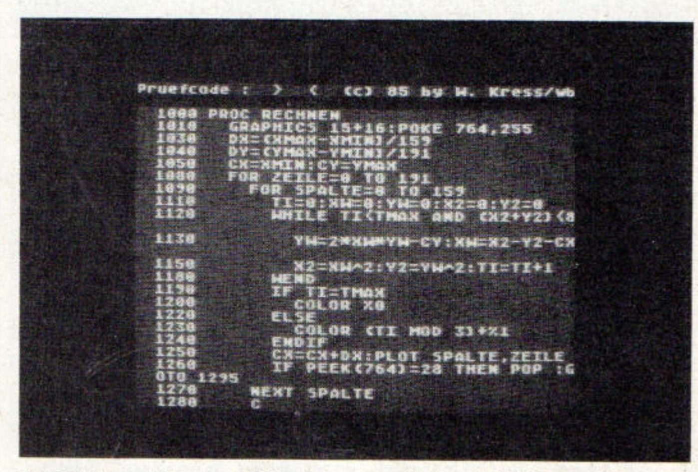


Bild 2. So sollte der Bildschirm nach dem Aktivieren des Prüfsummers aussehen.

»Prüfcode: > < (c) by W. Kress/wb«. Wie die Aufteilung des Bildschirms unter Verwendung des Prüfsummenprogramms aussieht, zeigt Bild 2.

2. Das Maschinenprogramm wird als Boot-File auf Kassette geschrieben. Das spätere Laden oder Booten geht folgendermaßen vor sich:
- bei ausgeschaltetem Computer die START-Taste drücken und Computer einschalten.
- nach Ertönen des Summtons die START-Taste loslassen
- nacheinander die PLAY-Taste des Kassettenrecorders und die RETURN-Taste betätigen.
- das Prüfsummenprogramm wird geladen und automatisch gestartet

3. Programm beenden. Das Maschinenprogramm wird nicht gespeichert.

PROGRAMM-STECKBRIEF	
Programmname	Prüfsummer
Programmtyp	Eingabehilfe
Programmiersprache	Atari-Basic und Maschinensprache
Programmlänge	4075 Byte
für Computer	alle
zusätzliche Hardware	Diskettenstation oder Kassettenrecorder
Eingabehilfe	keine
Bemerkung	Eingabehilfe für Atari-Basic und Turbo-Basic-XL-Listings.
Leserservice	Diskette (PRUEFSUM.BAS)

Das Prüfsummenprogramm wird in der Page 6 abgelegt, einem vor NEW geschützten Speicherbereich. Abschalten kann man es durch Drücken von SYSTEM RESET, ein Neustart erfolgt mit dem Aufruf: »?USR(1536)«.

Sobald das Prüfsummenprogramm aktiviert ist, können beim Zugriff auf ein Speichermedium Bildstörungen auftreten. Um dies zu vermeiden, brauchen Sie nach dem Laden oder nach dem Aufruf mit »?USR(1536)« nur »POKE 39998,30« einzugeben.

Wurde das Programm von Kassette gebootet, wird es jedesmal nach Betätigung der SYSTEM RESET-Taste erneut gestartet. Gibt man jedoch vor der Betätigung der SYSTEM RESET-Taste »POKE 9,0« ein, kann man diesen Effekt umgehen.

Bevor Sie nun ein eingetipptes Basic- oder Turbo-Basic XL-Programm mit RUN starten, sollten Sie es sicherheitshalber noch einmal speichern. Weiterhin sollte das Prüfsummen-Programm mit SYSTEM RESET desaktiviert werden, da auch manche Basic-Programme die Page 6 für Maschinenunterprogramme verwenden. Es könnte sonst zu Überschneidungen und zum Systemabsturz kommen.

Der Prüfcode besteht jeweils aus zwei Zeichen. Er setzt sich aus den Buchstaben von A bis einschließlich Z zusammen. Geben Sie nun probeweise eine Programmzeile ein und schließen Sie mit der RETURN-Taste ab. Der zugehörige Prüfcode erscheint dann in der obersten Zeile. Sollte der Code nicht mit dem im abgedruckten Listing übereinstimmen, kann die Programmzeile sofort verbessert werden. Die jeweilige Programmzeile braucht dazu nicht erneut eingegeben werden, da sämtliche Editierfunktionen des Atari-Computers zur Verfügung stehen.

Ein wichtiger Hinweis:
Für die erste Programmzeile kann eventuell ein falscher Prüfcode errechnet werden. Sollte die Prüfsumme also nicht mit der im Listing abgedruckten übereinstimmen, fahren Sie mit dem Cursor nochmals auf die erste Zeile. Sobald Sie die RETURN-Taste betätigt haben, erscheint der richtige Code auf dem Bildschirm. Stimmt der ausgegebene Wert dann immer noch nicht, überprüfen Sie nochmals jedes einzelne Zeichen der von Ihnen eingegebenen Programmzeile.

Bei der Eingabe von Programmen sind die üblichen Abkürzungen für Basic-Befehle zulässig. Sie können also zum Beispiel G. für GOTO, SE. für SETCOLOR und so weiter verwenden. Eine Ausnahme gilt jedoch bei der Abkürzung für den PRINT-Befehl, den man in der Regel mit einem Fragezeichen abkürzen kann. **Verwenden Sie das Fragezeichen bitte nur dann, wenn es auch im Listing verwendet wird.**

Möchten Sie nachträglich ein Programm überprüfen, so laden Sie zuerst das Prüfsummenprogramm und starten es mit »?USR(1536)«. Anschließend laden Sie Ihr Programm und listen einen Teil auf dem Bildschirm. Dann bewegen Sie den Cursor an irgendeine Stelle einer Programmzeile. Mit

Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)
A	B	C	A	B	C	A	B	C	A	B	C
0		CTL ,	19		CTL S	128		(Invers) CTL ,	147		(Invers) CTL S
1		CTL A	20		CTL T	129		(Invers) CTL A	148		(Invers) CTL T
2		CTL B	21		CTL U	130		(Invers) CTL B	149		(Invers) CTL U
3		CTL C	22		CTL V	131		(Invers) CTL C	150		(Invers) CTL V
4		CTL D	23		CTL W	132		(Invers) CTL D	151		(Invers) CTL W
5		CTL E	24		CTL X	133		(Invers) CTL E	152		(Invers) CTL X
6		CTL F	25		CTL Y	134		(Invers) CTL F	153		(Invers) CTL Y
7		CTL G	26		CTL Z	135		(Invers) CTL G	154		(Invers) CTL Z
8		CTL H	27		ESC ESC	136		(Invers) CTL H	156		ESC SHIFT DEL
9		CTL I	28		ESC CTL –	137		(Invers) CTL I	157		ESC SHIFT >
10		CTL J	29		ESC CTL =	138		(Invers) CTL J	158		ESC CTL TAB
11		CTL K	30		ESC CTL +	139		(Invers) CTL K	159		ESC SHIFT TAB
12		CTL L	31		ESC CTL *	140		(Invers) CTL L	224		(Invers) CTL .
13		CTL M	96		CTL .	141		(Invers) CTL M	251		(Invers) CTL ;
14		CTL N	123		CTL ;	142		(Invers) CTL N	252		(Invers) SHIFT =
15		CTL O	124		SHIFT =	143		(Invers) CTL O	253		ESC CTL 2
16		CTL P	125		ESC CTL <	144		(Invers) CTL P	254		ESC CTL DEL
17		CTL Q	126		ESC DEL	145		(Invers) CTL Q	255		ESC CTL >
18		CTL R	127		ESC TAB	146		(Invers) CTL R			

Tabelle der Atari-Grafikzeichen

Betätigung der RETURN-Taste wird der Prüfcode in der Statuszeile dargestellt. Stimmt der Prüfcode mit dem im beigefügten Listing überein, betätigen Sie wieder die RETURN-Taste, und der Prüfcode der nächsten Zeile wird errechnet und angezeigt. Auf diese Art und Weise können Sie ein komplettes Programm oder auch nur einzelne Zeilen »durchchecken«.

Sollte der Code nicht mit dem des Listings übereinstimmen, überprüfen Sie Ihre Programmzeile. Denn bei allen abgetippten Listings gilt: Die Programme müssen so abgetippt werden, wie sie abgedruckt sind. Stimmt die Anzahl der Leerzeichen? Haben Sie eventuell Zeichen vertauscht? Oder haben Sie ein »?« anstelle eines PRINT-Befehls eingegeben? Haben Sie den Fehler gefunden? Dann weiterhin viel Erfolg mit Ihren Atari-Programmen.

Besondere Hinweise zum Prüfsummenprogramm

Wenn Ihnen die Redaktion unseren Atari-Prüfsummer bereits zugeschickt hat, müssen Sie eine geringfügige Änderung am Basic-Programm vornehmen. In der alten Version funktioniert der Prüfsummer zwar bei der Eingabe von normalen Basic-Programmen, nicht aber bei Turbo-Basic XL-Programmen. Sie müssen aber lediglich am Ende von Zeile 630 ein Semikolon »;« anfügen. Sie sollte dann wie folgt aussehen:

```
630 IF D THEN ? #1;D$(1,500);
```

Starten Sie das Prüfsummenprogramm anschließend mit RUN und speichern Sie es wieder auf Ihrem Massenspeicher. Auf der Leserservice-Diskette befindet sich natürlich die neue, hier abgedruckte Prüfsummer-Version.

Wenn Sie ein Turbo-Basic XL-Programm eingeben möchten, müssen Sie das Prüfsummenprogramm mit einem Hilfsprogramm starten. Weiterhin empfiehlt es sich, eine separate Diskette, sozusagen als Eingabediskette zu verwenden. Formatieren Sie also eine Diskette. Verwenden Sie aber bitte **nur DOS 2.0 oder 2.5!** Speichern Sie anschließend die DOS-Files und Turbo-Basic XL mit dem Namen »AUTORUN.SYS« auf dieser Diskette. Anschließend schalten Sie den Computer aus und booten die soeben angelegte Diskette. Geben Sie anschließend folgende Programmzeile ein:

```
1 BLOAD "D:PRUEFER.OBJ":POKE $BC3E,$1E:*L-:NEW
```

Speichern Sie diese Programmzeile dann mit »SAVE" D:AUTORUN.BAS"« auf Ihrer Eingabediskette. Dieses kurze Programm sorgt beim Booten des Computers dafür, daß das Prüfsummenprogramm automatisch geladen und ausgeführt wird. Achtung: **Das Prüfsummenprogramm läßt sich nicht vom DOS-Menü aus starten! Die oben aufgeführte Programmzeile muß unbedingt beim Bootvorgang ausgeführt werden.** Auf Ihrer Prüfsummer-Diskette sollten sich also folgende Files befinden:

1. DOS.SYS (entweder DOS 2.0 oder 2.5)
2. DUP.SYS (passend zum DOS.SYS-File)
3. AUTORUN.SYS (Turbo-Basic XL)
4. AUTORUN.BAS (das Initialisierungsprogramm)
5. PRUEFER.OBJ (das eigentliche Prüfsummenprogramm als Maschinencode-File)

Basic-Programme schnell und zuverlässig eingegeben

Um also, unter Turbo-Basic XL, den Prüfsummer verwenden zu können, brauchen Sie nur Ihre Prüfsummer-Diskette ins Laufwerk einschieben und Ihren Computer einschalten. Alles weitere erledigt der Computer von selbst.

Vergessen Sie bitte nicht, sich von der neuen Prüfsummer-Version auch eine Sicherheitskopie anzulegen. Mit der alten Version sollten Sie unter keinen Umständen mehr arbeiten. Sind noch Fragen offen, wenden Sie sich bitte an die Redaktion. (wb)

```
100 REM ****************************     <LU>
110 REM *                          *     <HA>
120 REM *   PRUEFCODE fuer ATARI   *     <CU>
130 REM *                          *     <HE>
140 REM *  von  W.Kress            *     <ZZ>
150 REM *       Lindenweg 17       *     <XA>
160 REM *       7590 Achern        *     <IO>
170 REM *  (c)  by Happy-Computer  *     <DC>
180 REM *                          *     <HO>
190 REM ****************************     <MM>
200 REM                                  <AJ>
210 DIM F$(15),SED$(2),D$(504),C$(11)    <RC>
220 C$="{CTL ,}{CTL D}u7{CTL A}{CTL F})<
{CTL M}{CTL B}S"                         <IZ>
230 GRAPHICS 0:POKE 82,0:POKE 752,1      <KV>
240 POSITION 12,10:POKE 201,8            <BO>
250 ? "Bitte warten !":POKE 752,0        <DX>
260 Z=1000                               <GZ>
270 FOR I=1 TO 497 STEP 8                <OL>
280 PSUMM=0                              <MG>
290 FOR J=0 TO 7                         <PK>
300 READ SED$                            <ZV>
310 H=ASC(SED$(1))-48                    <EQ>
320 IF H>9 THEN H=H-7                    <LT>
330 L=ASC(SED$(2))-48                    <GE>
340 IF L>9 THEN L=L-7                    <SR>
350 DEZ=H*16+L:D$(I+J)=CHR$(DEZ)         <JV>
360 PSUMM=PSUMM+DEZ                      <MV>
370 NEXT J                               <GR>
380 READ ZSUMM                           <AW>
390 IF PSUMM=ZSUMM THEN 410              <IN>
400 ? "{ESC CTL 2}DATENFEHLER !":LIST Z:
END                                      <YX>
410 GSUMM=GSUMM+PSUMM:Z=Z+10             <NJ>
420 NEXT I                               <FY>
430 READ ESUMM                           <SL>
440 IF ESUMM=GSUMM THEN 460              <ZO>
450 ? "{ESC CTL 2}FEHLENDE ODER DOPPELTE
 DATA-ZEILE":END                         <IZ>
460 ? "{ESC CTL 2}"                      <EX>
470 ? "{ESC CTL <}"                      <RP>
480 POSITION 2,6:? :POKE 752,1           <UQ>
490 ? ,"{CTL Y}1{CTL Y} Auf DISKETTE sch
reiben{ESC CTL =}{ESC CTL =}"            <AY>
500 ? ,"{CTL Y}2{CTL Y} Auf CASSETTE sch
reiben{ESC CTL =}{ESC CTL =}"            <AQ>
510 ? ,"{CTL Y}3{CTL Y} ENDE"            <VR>
520 OPEN #1,4,0,"K":GET #1,K:CLOSE #1    <NQ>
530 K=K-48:IF K<1 OR K>3 THEN ? "{ESC CT
L 2}";:GOTO 520                          <WU>
540 POSITION 3,4+3*K:? "--->";           <FU>
550 POSITION 2,20:? "{ESC SHIFT DEL}";   <EW>
560 ON K GOTO 570,590,680                <DI>
570 POKE 752,0:? "{ESC TAB}DATEINAME :  
```

Listing zum Atari-Prüfsummer



```
D1:{ESC CTL +}{ESC CTL +}{ESC CTL +}{ESC
 CTL +}";                                        <KM>
580 POKE 764,44:INPUT F$:D=1:GOTO 610            <ZI>
590 F$="C:":D=0                                  <MT>
600 ? ,"CASSETTE▲FERTIG▲MACHEN,▲RETURN";         <AH>
610 TRAP 660                                     <OJ>
620 OPEN #1,8,128,F$                             <YW>
630 IF D THEN ? #1;D$(1,500);                    <UB>
640 IF  NOT D THEN ? #1;C$;D$(7,494)             <XK>
650 CLOSE #1:GOTO 470                            <EW>
660 ? "{ESC CTL 2}{ESC CTL -}{ESC SHIFT
DEL}{ESC TAB}I/O▲-▲FEHLER▲";PEEK(195);           <CR>
670 TRAP 40000:CLOSE #1:GOTO 480                 <HQ>
680 POKE 752,0:END                               <YW>
690 REM ----------------------------             <ZP>
1000 DATA▲FF,FF,00,38,E7,39,A2,FF,1271           <EA>
1010 DATA▲BD,EB,38,9D,00,06,CA,E0,1069           <NF>
1020 DATA▲FF,D0,F5,A2,60,BD,8A,38,1349           <VW>
1030 DATA▲9D,00,01,CA,10,F7,A2,0F,800            <OL>
1040 DATA▲BD,00,E4,9D,61,01,CA,10,890            <LI>
1050 DATA▲F7,18,AD,65,01,69,01,8D,793            <MP>
1060 DATA▲6B,06,AD,66,01,69,00,8D,635            <CE>
1070 DATA▲6C,06,A9,2A,8D,65,01,A9,737            <XW>
1080 DATA▲06,8D,66,01,AD,67,01,8D,668            <VG>
1090 DATA▲30,06,AD,68,01,8D,34,06,531            <HH>
1100 DATA▲38,AD,30,02,E9,02,8D,02,657            <ZR>
1110 DATA▲06,8D,11,06,AD,31,02,E9,627            <EV>
1120 DATA▲00,8D,07,06,8D,12,06,A9,488            <DM>
1130 DATA▲45,20,7A,38,90,11,8E,26,620            <CP>
1140 DATA▲06,A9,00,20,7A,38,90,07,536            <YQ>
1150 DATA▲8E,17,06,20,01,06,18,60,330            <QZ>
1160 DATA▲A2,00,DD,1A,03,F0,08,E8,892            <CH>
1170 DATA▲E8,E8,E0,21,D0,F4,18,60,1293           <CK>
1180 DATA▲70,42,39,01,9E,06,9E,06,564            <UJ>
1190 DATA▲9E,06,75,06,9E,06,9E,06,615            <EX>
1200 DATA▲00,00,00,00,4C,2E,22,54,240            <CY>
1210 DATA▲3A,22,2C,50,45,45,4B,28,469            <TN>
1220 DATA▲31,31,33,29,2B,32,35,36,390            <LL>
1230 DATA▲2A,50,45,45,4B,28,31,31,473            <SG>
1240 DATA▲34,29,9B,01,61,45,01,04,420            <SW>
1250 DATA▲54,30,72,75,65,66,63,6F,776            <YW>
1260 DATA▲64,65,00,1A,00,00,1E,00,257            <ER>
1270 DATA▲00,1C,00,00,08,63,09,00,144            <LV>
1280 DATA▲18,15,00,62,79,00,37,0E,333            <NU>
1290 DATA▲00,2B,72,65,73,73,0F,77,622            <KV>
1300 DATA▲62,68,A9,FF,8D,30,02,A9,986            <OK>
1310 DATA▲FF,8D,31,02,A0,03,B9,00,795            <ZL>
1320 DATA▲01,99,FF,FF,88,10,F7,A0,1223           <DR>
1330 DATA▲FF,A2,05,BD,33,01,99,1A,842            <OO>
1340 DATA▲03,C8,E0,03,D0,02,A0,FF,1055           <QQ>
1350 DATA▲CA,10,F0,60,A5,77,F0,0A,1088           <OT>
1360 DATA▲A9,FF,85,26,A9,FF,85,27,1191           <LE>
1370 DATA▲C6,77,A5,73,F0,2D,A5,A1,1208           <VB>
1380 DATA▲10,04,C6,73,F0,25,A6,78,896            <NG>
1390 DATA▲BD,14,01,E6,78,C9,9B,D0,1124           <ZM>
1400 DATA▲4F,A2,00,86,78,86,73,A6,910            <WM>
1410 DATA▲A0,86,71,A6,A1,86,72,A2,1144           <QR>
1420 DATA▲9E,86,26,A2,06,86,27,E6,901            <AY>
1430 DATA▲77,D0,35,20,3E,F6,08,C9,929            <CM>
1440 DATA▲9B,D0,02,E6,73,28,60,A2,1008           <IO>
1450 DATA▲00,C9,9B,F0,26,86,CC,A6,1138           <LN>
1460 DATA▲00,86,D1,85,CD,A0,08,46,919            <OZ>
1470 DATA▲D1,90,0C,18,A2,03,B5,CD,940            <DL>
1480 DATA▲75,CA,95,CD,CA,D0,F7,06,1336           <ZZ>
1490 DATA▲CD,26,CC,88,D0,E9,E6,00,1254           <EW>
1500 DATA▲A0,01,60,A9,A4,A0,02,20,784            <HA>
1510 DATA▲CC,06,A2,03,B5,CA,95,CD,1112           <MP>
1520 DATA▲CA,D0,F9,A9,1A,20,CC,06,1096           <IN>
1530 DATA▲A5,CD,18,69,21,8D,48,01,746            <BX>
1540 DATA▲A5,D0,69,21,8D,47,01,84,856            <UQ>
1550 DATA▲D0,C8,84,00,60,85,72,84,1015           <LY>
1560 DATA▲71,86,CB,86,CC,86,CD,A0,1287           <JS>
1570 DATA▲18,18,A2,05,36,CB,CA,10,690            <RJ>
1580 DATA▲FB,38,A2,03,B5,CA,F5,6F,1211           <RY>
1590 DATA▲95,73,CA,D0,F7,90,0B,B5,1257           <GI>
1600 DATA▲74,95,CB,E8,E0,03,D0,F7,1382           <II>
1610 DATA▲E6,D0,88,D0,DC,60,E2,02,1326           <JV>
1620 DATA▲E3,02,00,38,00,00,00,00,285            <BR>
1630 DATA▲53423                                  <NQ>
```

Listing zum Atari-Prüfsummer (Schluß)

Farbe auf die Fläche

Das normale Atari-Basic kennt leider keinen Befehl, eine Fläche einzufärben. Das XIO-Kommando ist nur ein schlechter Ersatz. Das Programm »Fill« bringt Farbe auf die Fläche und schraffiert sie sogar.

Das Maschinensprache-Programm »Fill« besteht aus zwei Teilen:

1) Dem Hauptprogramm (333 Byte lang), das in einen String geladen werden muß (beim Demoprogramm MA$).

2) Den Unterpgrogrammen (86 Byte lang), die in Page 6 von der Adresse 1536 bis 1621 stehen.

Bevor »Fill« gestartet werden kann, müssen die Zahlen aus den DATA-Zeilen geladen werden. Nach dem Einschalten von »Grafik 8« wird die Figur gezeichnet, die ausgefüllt werden soll. Um die Fläche festzulegen, wird mit »COLOR 0« und »PLOT X,Y« ein Punkt auf die Stelle gesetzt, von der aus das Fill-Programm starten soll. COLOR 0 ist deshalb wichtig, damit der Punkt unsichtbar bleibt und beim Ausfüllen nicht als Hindernis erkannt wird.

Danach wird das »Fill«-Programm mit »Q=USR(ADR(MA$), MUSTER)« gestartet. »ADR(MA$)« gibt die Anfangsadresse von MA$ im RAM-Speicher und damit auch die Startadresse des Maschinensprache-Programms an. »MUSTER« kann eine Variable oder einen Wert zwischen 0 und 255 sein. Hiermit wird das Bit-Muster angegeben, mit dem ausgefüllt werden soll. Der Dezimalwert 255 ergibt als Binärzahl 11111111, die sich wie folgt errechnet:

$1x2^0+1x2^1+1x2^2...+1x2^7=255$.

Hier wird jeder Punkt gezeichnet. Bei Dezimalwert 170 (entspricht binär 10101010) wird nur jeder zweite Punkt auf dem Bildschirm sichtbar. Das Bitmuster wird in jeder Bildschirmzeile um einen Punkt verschoben, so daß sich daraus schräg schraffierte Flächen (von links oben nach rechts unten) ergeben. Allerdings muß die Variable »MUSTER« dann ungleich 0 sein.

Um die Funktion des Programmes zu verstehen, ist es wichtig, etwas über den Bildschirmspeicher in Grafikstufe 8 zu wissen. Die Anfangsadresse des Bildschirmspeichers steht in den Adressen 88 und 89. Sie wird von Basic aus mit »PRINT PEEK(88)+PEEK(89)*256« errechnet.

In Grafikstufe 8 ist der Bildschirmspeicher aus 192 Zeilen mit je 40 Byte aufgebaut. Jede Zeile ist auf dem Bildschirm einen Punkt hoch. Ein gesetztes Bit aus einem Byte bedeu-

PROGRAMM-STECKBRIEF	
Programmname	Fill-Routine
Programmtyp	Grafik-Utility
Programmiersprache	Basic und Maschinensprache
Programmlänge	3365 Byte
für Computer	600XL/800XL/130XE
zusätzliche Hardware	keine
Eingabehilfe	Prüfsummer
Bemerkung	zum Füllen von Flächen in den Grafikstufen 4, 6 und 8
Leserservice	Diskette (FILL.BAS)

tet, der Punkt ist sichtbar. Pro Byte können also acht Punkte gespeichert werden. Bei 40 Byte pro Zeile mit je acht Punkten kommt man wieder auf 40 x 8 = 320 Punkte Auflösung. Schreibt man in die erste Adresse den Wert 128, erscheint links oben ein Punkt. Addiert man zu der Adresse 40, wird der Punkt eine Zeile tiefer gezeichnet.

Das Programm »Fill« errechnet zuerst aus der Grafikcursorposition, die durch »PLOT« gesetzt wird (Adresse 90 für vertikale und 91/92 für horizontale Position), die Speicheradresse im Bildschirmspeicher. Dieser Wert wird in XA, YA und XS, YS gespeichert (in Maschinensprache gibt es keine Variablen; hier müssen freie Speicherzellen verwendet werden). XA und YA sind die Koordinaten des Arbeitspunktes, der durch das Programm verschoben wird. XS und YS werden, nachdem die erste Hälfte der Fläche ausgefüllt ist, dazu verwendet, den Arbeitspunkt wieder auf die Ausgangsposition zu setzen.

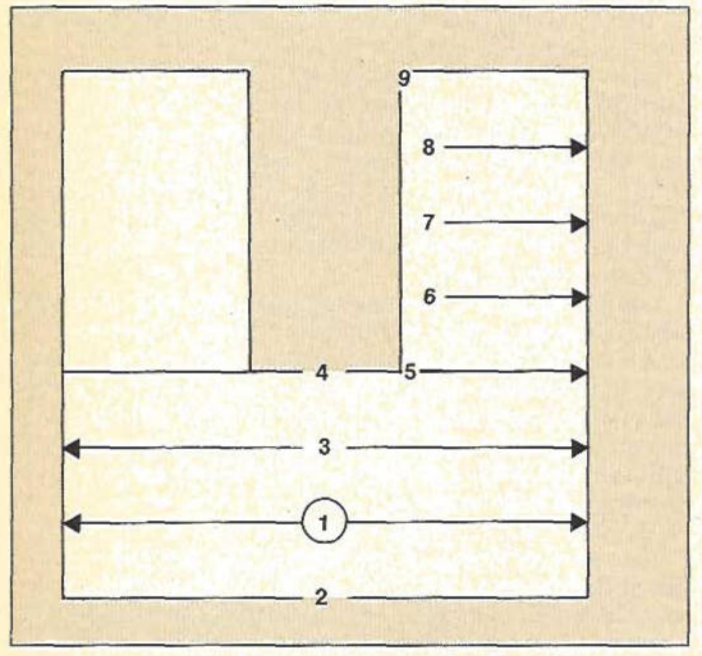


Bild 1. Verschiebung des Arbeitspunktes beim Ausmalen einer Fläche. Bei diesem Beispiel erfolgt das Füllen der Fläche rechtsbündig. Um auch den linken Teil füllen zu können, muß der Arbeitspunkt in die noch leere Fläche gesetzt werden und das Programm»Fill« nochmals gestartet werden.

Vom Arbeitspunkt aus zeichnet das Programm so lange eine Linie nach rechts, bis es auf ein Hindernis stößt. In Maschinensprache heißt das, daß von der Position XA, YA aus jedes Bit im Bildschirmspeicher abgefragt werden muß. Wird bei der Abfrage ein bereits gesetztes Bit gefunden, stellt dieses den Rand der zu füllenden Fläche dar. Der Füllvorgang wird dann in umgekehrter Richtung fortgesetzt. Das gleiche gilt, wenn der Bildschirmrand erreicht ist. Findet das Programm auch hier ein Hindernis, wird YA um 1 erhöht (der Arbeitspunkt wird um eine Zeile nach unten verschoben). Ist ein Hindernis direkt auf dem Arbeitspunkt XA, YA, kann weder eine Linie nach rechts noch nach links gezeichnet werden. Der Arbeitspunkt muß dann so weit nach rechts verschoben werden, bis eine freie Stelle auf dem Bildschirm gefunden ist. Auch hierfür werden wieder die einzelnen Bits im Bildschirmspeicher getestet. Liegt der neue Arbeitspunkt noch innerhalb der auszumalenden Fläche, setzt sich das Ausfüllen der Fläche fort.

Liegt der neue Arbeitspunkt aber schon außerhalb der auszumalenden Fläche, wird er so lange nach links verschoben, bis eine freie Stelle gefunden wird. Bild 1 verdeutlicht diesen

XS	= 90	$5A	
XS	= 91	$5B	Startposition
Bit	= 92	$5C	
YA	= 96	$60	
XA	= 97	$61	Arbeitspunkt
BitA	= 98	$62	
Rand R	= 100	$64	rechter Rand der Fläche
Rand RB	= 101	$65	
Rand L	= 102	$66	linker Rand der Fläche
Rand LB	= 103	$67	
Lücke	= 104	$68	
Sicher	= 105	$68	kurzzeitiger Speicher für Zwischenergebnisse
Modus	= 107	$6B	
Bild	= 108, 109,	$6C, 6D	Adressiert die Stelle im Bildschirmspeicher, an der gerade gearbeitet wird
Muster	= 110	$6E	wird über den USR-Befehl gesetzt

Bild 2. Verwendete Adressen

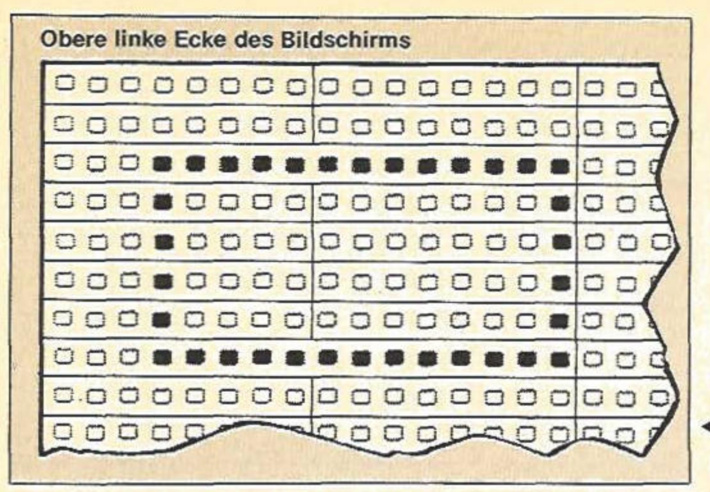
Vorgang. Ist auch sie schon außerhalb der auszufüllenden Fläche, ist deren unteres Ende erreicht. Der Arbeitspunkt wird jetzt wieder auf den Ausgangspunkt gesetzt (XA=XS; YA=YS). Dann wird die Fläche wieder ausgemalt, nur das XA nach jeder fertigen Zeile nicht um 1 erhöht, sondern erniedrigt (eine Bildschirmzeile nach oben). Kann auch hier der Arbeitspunkt nicht mehr innerhalb der auszumalenden Fläche verschoben werden, oder ist der obere Bildschirmrand erreicht, wird ins Basic zurückgekehrt.

Mit der »Fill«-Routine können fast alle Flächen ausgefüllt werden. Teilt sie sich aber in vertikaler Richtung (zum Beispiel bei einer U-Form wie in Bild 1), wird nur einer der Arme ausgefüllt. Hier muß der Ausgangspunkt mit »COLOR 0« und »PLOT X,Y« in den noch nicht ausgemalten Bereich gesetzt werden und das Programm »Fill« muß noch einmal gestartet werden.

Eine Fläche mit mehreren Verzweigungen in horizontaler Richtung wird jedoch schon beim ersten Mal korrekt ausgefüllt. Das Programm arbeitet normalerweise nur in Grafikstufe 8. Es kann aber auch für Stufe 6 oder 4 umgeändert werden, da sie fast den selben Bildschirmspeicheraufbau besitzen. Dafür müssen folgende POKE-Befehle an einer Stelle nach dem Einlesen von »Fill« aus den DATA-Zeilen in das Basic-Programm gesetzt werden: A=ADR(MA$)

Grafikstufe	4	6	8
POKE A+71,	48	96	192
POKE A+153	9	19	39
POKE A+260,	9	19	39
POKE 1552,	10	20	40

Positionen werden bei »Fill« in einer 1-Byte-Adresse für die vertikale (y-Achse) und in einer 2-Byte-Adresse für die horizontale Richtung gespeichert. So bekommt man mit »PRINT PEEK(84)« die momentane Zeile und mit »PRINT PEEK(85)+(PEEK(86)*256« die momentane Spalte des Grafikcursors. Mit »PRINT PEEK(90)« erhält man die vorhergehende Zeile und mit »PRINT PEEK(91)+PEEK (92)*256« die vorhergehende Spalte, in der sich der Grafikcursor befunden hat. Diese Adressen, die als dezimaler Byte-Wert gePEEKt werden, werden auch von den Basic-Befehlen DRAWTO und XIO 18 (Füllen von Flächen) verwendet. In Bild 2 sind noch alle von »Fill« verwendeten Adressen aufgeführt. Bild 3 und Bild 4 verdeutlichen noch den Aufbau des Grafikbildschirms in Stufe 8. Die einzelnen Bildschirmdaten werden Zeile für Zeile im RAM-Speicher abgelegt. Um die Startadresse des Bildschirmspeichers zu bestimmen, müssen die



◀ **Bild 3. Ausschnitt des Bildschirms in Grafikstufe 8. Die ausgefüllten Punkte sind sichtbar, alle anderen Punkte — auch Pixel genannt — sind unsichtbar.**

Adressen 88 und 89 abgefragt werden. Geben Sie hierzu bitte folgende Zeile ein: »PRINT PEEK(88)+PEEK(89) *256«.

Das Programm benützt zwei Koordinatenpaare zum Speichern von Positionen auf dem Bildschirm:

XA, YA = Arbeitspunkt: Dieser Punkt wird vom Programm geändert. Von hier aus werden im Programm Linien nach rechts und nach links gezeichnet.

XS, YS = Startposition: Dieser Punkt wird nicht geändert und zum Setzen von XA und YA verwendet. Die Startposition errechnet sich aus der Position des Grafikcursors.

(Rolf Kilian/wb)

Byte 0 Byte 1 Byte 2 Byte 3 Byte 4 Byte 5 Byte 6 Byte 7

Oberste Bildschirmzeile (Zeile 0) und so weiter

Byte 40 Byte 41 Byte 42 Byte 43 Byte 44 Byte 45 Byte 46 Byte 47

und so weiter

Bild 4. Jede Bildschirmzeile in Grafikstufe 8 setzt sich aus 40 Bytes mit jeweils 8 Bits zusammen. Daraus ergibt sich die horizontale Auflösung von 320 Punkten. Der Bildschirm setzt sich aus 192 solcher Zeilen zusammen.

```
10 REM Demoprogramm zu "FILL"                  <RT>
20 REM ***************                         <JB>
30 REM ROLF KILIAN                             <HX>
40 REM Oeschwende 11                           <FQ>
50 REM 7900 Ulm-Lehr                           <CL>
60 REM Tel. 0731/60591                         <WZ>
70 REM ***************                         <JG>
80 REM Oktober 1984                            <ON>
90 REM                                         <JW>
100 DIM MA$(333):REM Das Hauptprogramm w
ird in MA$ gespeichert                         <HR>
110 DEG                                        <WC>
120 A=ADR(MA$):IF PEEK(A)=104 AND PEEK(A
+332)=166 THEN GOTO 200                        <QC>
125 ? CHR$(125);"BITTE WARTEN"                 <GT>
130 REM *** Zeile 140 kann nach dem 1.fe
hlerfreien Lauf geloescht werden               <XS>
140 RESTORE 30000:FOR I=0 TO 418:READ P:
C=C+P:NEXT I:IF C<>50795 THEN ? "FEHLER 
IN DATAZEILEN":END                             <FC>
150 RESTORE 30000                              <BD>
160 FOR I=0 TO 332:READ P:POKE ADR(MA$)+
I,P:NEXT I:REM *** Hauptprogramm einlese
n                                              <UA>
170 RESTORE 30046                              <HR>
180 FOR I=0 TO 85:READ P:POKE 1536+I,P:N
EXT I:REM *** Unterprogramme einlesen          <LY>
199 REM *** Pyramiede aus Kreisen zeichn
en                                             <BJ>
200 A=0:GRAPHICS 8+16                          <LL>
210 R=20:S=30                                  <RP>
220 FOR YP=30 TO 150 STEP 30                   <XG>
230 FOR XP=175-YP/2 TO 145+YP/2 STEP 30        <JM>
240 COLOR 1:GOSUB 500:REM *** Kreis zeic
hnen                                           <GR>
250 COLOR 0:PLOT XP,YP:REM *** Ausgangsp
osition fuer Fill                              <GI>
260 Q=USR(ADR(MA$),RND(0)*256):REM *** F
ill mit zufaelligem Muster starten             <RH>
270 NEXT XP                                    <TW>
280 NEXT YP                                    <UI>
290 R=80:YP=95:XP=160:S=15                     <VA>
300 COLOR 1:GOSUB 500                          <SE>
310 COLOR 0:PLOT 100,100:Q=USR(ADR(MA$),
255)                                           <VA>
320 PLOT 220,100:Q=USR(ADR(MA$),255):REM
 *** Starten von Fill; Muster=255 -> Fla
eche ganz ausfuellen                           <TS>
330 PLOT 175,165:Q=USR(ADR(MA$),255)           <PU>
340 PLOT 145,165:Q=USR(ADR(MA$),255)           <OP>
350 R=95                                       <AA>
360 COLOR 1:GOSUB 500                          <SQ>
370 COLOR 0:PLOT 50,100:Q=USR(ADR(MA$),1
70)                                            <FG>
380 PLOT 300,100:Q=USR(ADR(MA$),170)           <ZT>
390 GOTO 390                                   <QP>
499 REM *** KREIS ZEICHNEN                      <QT>
500 FOR ALFA=0 TO 360 STEP S                   <UB>
510 X=XP+COS(ALFA)*R:Y=YP+SIN(ALFA)*R          <XQ>
520 IF ALFA=0 THEN PLOT X,Y                    <TF>
530 DRAWTO X,Y                                 <QG>
540 NEXT ALFA                                  <WA>
550 RETURN                                     <MZ>
30000 DATA 104,133,105,104,104,133,110,7
2,162,3,102,92,102,91,102                      <YY>
30002 DATA 105,202,208,247,162,4,38,105,
202,208,251,165,105,133,92                     <TH>
30004 DATA 133,98,133,103,133,101,165,91
,133,97,133,102,133,100,165                    <ZH>
30006 DATA 90,133,96,169,1,133,107,208,7
1,176,18,169,0,24,102                          <RB>
30008 DATA 110,106,5,110,133,110,230,96,
165,96,201,192,208,234,104                     <PN>
30010 DATA 133,110,165,92,133,98,133,103
,133,101,165,91,133,97,133                     <EF>
30012 DATA 102,133,100,165,90,133,96,169
,0,133,107,165,96,201,0                        <JP>
30014 DATA 240,17,198,96,169,0,6,110,42,
5,110,133,110,208,5                            <PV>
30016 DATA 176,2,144,233,96,32,0,6,166,9
8,32,54,6,133,104                              <IU>
30018 DATA 164,97,49,108,162,0,42,176,28
,232,224,8,208,248,32                          <JQ>
30020 DATA 39,6,192,39,240,30,200,177,10
8,208,234,32,39,6,208                          <YH>
30022 DATA 242,165,107,208,140,240,204,1
96,97,208,4,228,98,240,65                      <TR>
30024 DATA 32,70,6,32,41,6,134,101,132,1
00,166,98,32,70,6                              <RD>
30026 DATA 133,104,164,97,49,108,162,8,1
```

Basic-Listing zu »Fill«

```
06,202,176,19,208,250,32                    <VC>
30028 DATA 39,6,192,0,240,16,136,177,108
,208,236,32,39,6,208                        <XK>
30030 DATA 242,32,54,6,32,41,6,134,103,1
32,102,24,232,208,182                       <MP>
30032 DATA 56,176,250,144,141,224,7,240,
6,42,232,144,17,176,246                     <CN>
30034 DATA 196,100,176,26,192,39,240,22,
200,177,108,162,255,208,231                 <BT>
30036 DATA 196,100,208,4,228,101,176,7,1
32,97,134,98,24,144,214                     <NQ>
30038 DATA 166,98,32,54,6,164,97,17,108,
162,8,106,202,144,16                        <OY>
30040 DATA 208,250,192,0,240,190,136,196
,102,144,185,177,108,56,176                 <LW>
30042 DATA 234,196,102,208,6,228,103,240
,172,144,170,132,97,134,98                  <BL>
30044 DATA 24,144,166                       <SZ>
30046 DATA 165,96,133,105,169,0,133,108,
162,8,70,105,144,3,24                       <CW>
30048 DATA 105,40,106,102,108,202,208,24
3,133,109,24,165,108,101,88                 <QT>
30050 DATA 133,108,165,109,101,89,133,10
9,96,169,255,37,104,37,110                  <VX>
30052 DATA 17,108,145,108,169,255,133,10
4,96,169,255,224,0,240,9                    <FC>
30054 DATA 134,105,24,106,202,208,251,16
6,105,96,169,0,224,0,240                    <TM>
30056 DATA 249,134,105,56,106,202,208,25
1,166,105,96                                <UC>
```

Basic-Listing zu »Fill« (Schluß)

Mit »Happy-Mon« auf der Suche

Will man den RAM-Speicher durchforsten, sollte man einen Maschinensprache-Monitor verwenden. »Happy-Mon« ist sehr leistungsfähig und bietet alle Funktionen, die Sie brauchen.

Bekanntlich dient der RAM-Speicher eines Computers dazu, Werte hineinzuschreiben und wieder zu lesen. Diese Eigenschaft nutzt auch der eingebaute Basic-Interpreter im Atari-Computer. Allerdings fällt es schwer, von Basic aus zu verfolgen, welche Werte wie und wo im Speicher abgelegt wurden. Mit dem Basic-Befehl PEEK kann man seine Neugierde zumindest teilweise befriedigen. Dabei erhält man einen Dezimalwert zwischen 0 und 255. Was fängt man aber damit an? Möchte man nämlich wissen, welchen Befehl der gePEEKte Wert ausführt, muß man eine Befehlscode-Tabelle zur Hand nehmen. Dieses Verfahren ist sehr aufwendig, und nicht selten schleichen sich noch Fehler ein. Hier ist »Happy-Mon« eine willkommene Hilfe, denn unseren Maschinensprache-Monitor lädt man einfach zusätzlich in den Speicher. Das Programm kann dann einen bestimmten Adressenbereich disassemblieren oder einen Hexadezimal- oder Dezimal-Dump auf dem Bildschirm ausgeben. Tabelle 1 zeigt eine Gesamtübersicht der verfügbaren Befehle von »Happy-Mon«.

Hexdump- und Editierfunktionen

Der Befehl »Z Adresse« bewirkt die Ausgabe eines Hexdumps ab Adresse »Adresse«. Es sind beliebige Anfangsadressen zwischen dezimal 0 und 65535 zulässig. Bevorzugen Sie hexadezimale Zahlen, müssen Sie das Kennzeichen $ verwenden. Dies gilt übrigens für alle numerischen Befehlsparameter.

Beispiel: »Z $8000« gibt die ersten acht Byte des »Happy-Mon«-Objektcodes in hexadezimaler Darstellung aus. Anschließend erscheint die Frage »Continue (Y/N)« wobei Y für »ja« steht. Betätigen Sie die Y- oder die RETURN-Taste, um die nächsten acht Byte auszugeben. Mit N für nein oder der BREAK-Taste übergibt man die Kontrolle wieder an den Eingabeeditor. Empfinden Sie die ständige Frage »Continue (Y/N)« als überflüssig, fügen Sie nur ein N an den Befehl an. Beispiel: Mit »Z $8000,N« erfolgt eine ständige Ausgabe, die nur mit CONTROL 1 oder mit der BREAK-Taste unterbrochen werden kann.

Übrigens können Befehlszeilen beliebig viele Leerzeichen enthalten. Der Befehl »Z $8000,N« hat dieselbe Funktion wie »Z $8000 , N«. Die einzige Einschränkung: Zwei Parameter dürfen nur von einem Komma getrennt sein.

Ähnlich den oben beschriebenen Befehlen arbeitet auch der L-Befehl. Die Syntax ist identisch zum Z-Befehl. Zusätzlich wird nur noch die ASCII-Umsetzung der Byte ausgegeben.

Die bisherigen Funktionen dienen nur zum Lesen bestimmter Adressen. Mit »Happy-Mon« sind aber auch verschiedene Speicherbereiche veränderbar. Dazu wird bei der Ausgabe von Speicherinhalten vor jede Zeile das Größer-Zeichen (>) gesetzt. Dieses Zeichen stellt im Eingabemodus wiederum einen Befehl dar. Erhält nämlich der Editor eine Zeile mit vorgestelltem Größer-Zeichen, so werden die acht darauffolgenden Werte in den RAM-Speicher übernommen. Eine erfolgreiche Übernahme bestätigt »Happy-Mon« mit einem kurzen, tiefen Ton.

Wurde also ein bestimmter Speicherbereich mit Z oder L gedumpt, können Sie diesen mit den üblichen Editierfunktionen wie DELETE, BACKSPACE oder INSERT verändern. Anschließend kann die betreffende Zeile mit der RETURN-Taste in den Speicher übernommen werden. Versuchen Sie diese Funktion aber bitte nicht an Adressen, die »Happy-Mon« belegt.

Beim Editieren muß sich übrigens mindestens eine Hexadezimalzahl hinter dem Doppelpunkt befinden. Bedenken Sie aber, daß sich eine Hexadezimalzahl stets aus zwei Zeichen zusammensetzt (0 bis 9 und A bis F). Die Verwendung und Anzahl von Leerzeichen ist freigestellt.

Weitere Editierfunktionen

Eine besonders schnelle Eingabe von größeren Hexcode-Blöcken erlaubt die Funktion A. Der Befehl »A $5000« bewirkt die Ausgabe von »>5000:« auf dem Bildschirm, wobei der Cursor hinter dem Doppelpunkt positioniert ist. Jetzt können bis zu acht Hexadezimalwerte eingegeben werden. Mit RETURN erfolgt die Übergabe der kompletten Zeile in den RAM-Speicher, und die nächste Anfangsadresse wird ausgegeben. Mit BREAK verläßt man diesen Modus wieder.

Ähnlich arbeitet Funktion E. Zusätzlich werden hier die ASCII-Werte ausgegeben. Mit RETURN ist eine Zeile dann im Speicher.

Der Disassembler

Zum Disassemblieren stehen »Happy-Mon« zwei verschiedene Modi zur Verfügung. Den ersten Modus ruft der Befehl »D Adresse(,N)« auf. Hier erfolgt die Ausgabe in mnemonischer Form und in der Reihenfolge: Adresse, Hexcode und

Befehl	Erklärung
Monitorfunktionen	
<hexnum: h1 bis h8	Editieren im Speicher
L Anfangsadresse (,N)	Hex- und ASCII-Dump
Z Anfangsadresse (,N)	Hexdump
A Anfangsadresse	Automatische Adressenausgabe
E Anfangsadresse	Automatische Adressen- und Dumpausgabe
Disassembler	
D Anfangsadresse (,N)	Disassemblieren
U Anfangsadresse (,N)	Erzeugen eines vom Assembler verstandenen Formats
Speicherzugriffe	
M Anfangsadresse, Neuanfang, Anzahl	Verschieben eines Speicherblocks
P Adresse, Byte	Single Byte POKE
V Adresse, Word	2 Byte-POKE (Low/High)
F Anfangsadresse, Anzahl, Byte	Füllen eines Speicherbereichs
T Anfangsadresse, "string"	Setzt einen String im ASCII-Format ein
Suchen im Speicher	
HB Anfangsadresse, Byte (,N)	Sucht ein Byte
HW Anfangsadresse, Word (,N)	Sucht einen 2-Byte-Wert (Low/High)
Ein- und Ausgabe	
O #Filespezifikation	setzt beziehungsweise ändert den zweiten Ausgabekanal
N	Löscht den zweiten Ausgabekanal
R Sektornummer, Bufferanfang	Liest einen Sektor von Diskette
W Sektornummer, Bufferanfang	Schreibt einen Sektor auf Diskette
Hexadezimal-Dezimal-Wandlung	
? Dezimalwert	Wandelt eine Dezimalzahl in eine Hexadezimalzahl um
? $Hexadezimalzahl	Wandelt eine Hexadezimalzahl in eine Dezimalzahl um
»Happy-Mon« verlassen	
X	Rückkehr zum DOS
Q	Rückkehr zum Modul (falls vorhanden)
Verschiedenes	
G Adresse	Führt eine Maschinensprache-Unterroutine aus
K	Umschaltung zwischen normaler und invertierter Darstellung
* Wert	Setzt den Programm-Zähler
+ Wert	Setzt die Variable »+«

Befehlsübersicht von »Happy-Mon«

Mnemonic. Die Zeilen beginnen mit dem >-Zeichen, so daß sich die Hexadezimalwerte wieder editieren lassen.

Der zweite Disassembler-Modus wird mit »U Adresse(,N)« aufgerufen. In diesem Modus werden zu den mnemonischen Codes keine Adressen und Hexcodes, sondern Zeilennummern ausgegeben. Den erzeugten Code kann man dann mit einem Assembler, wie beispielsweise MAC/65, übernehmen. Bytes, die sich nicht disassemblieren lassen, werden

Fehlermeldung	Erklärung
1	Falscher Hexadezimal-Wert
2	Falscher Dezimal-Wert
3	Byte/Zeropage erwartet
4	Angegebene Anfangsadresse oder erreichte Arbeitsadresse liegt im »gefährlichen« Bereich von $D410 bis $D7FF
5	Falls bei der Eingabe einer bestimmten Anzahl eine Null eingegeben wird, tritt diese Fehlermeldung auf
6	Unbekannter Befehl
7	Fehlerhafte Syntax in einer Hexadezimal-Zeile
8	Tritt nicht auf
9	Falscher Parametertyp
128–255	Betriebssystem-Fehler

Die Fehlermeldungen von »Happy-Mon«

PROGRAMM-STECKBRIEF	
Programmname	Happy-Monitor
Programmtyp	Utility
Programmiersprache	Maschinensprache
Programmlänge	6407 Byte
für Computer	600 XL/800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	AMPEL
Bemerkung	Die abgedruckte Version läuft mit DOS 2.0 oder DOS 2.5. Für DOS-XL-Besitzer ist eine zweite Version auf der Leserservice-Diskette vorhanden
Leserservice	Diskette (MONITOR.COM/MONITOR1.COM)

als ».BYTE« dargestellt. Beachten Sie im Zusammenhang mit der U-Funktion auch den O-Befehl.

Spezielle Speicherzugriffe

Befehl	Erklärung
»M Anfangsadresse, Neuanfangsadresse, Anzahl«	bewegt einen Speicherblock
»P Adresse, Byte«	Einzelbyte-POKE
»V Adresse, Word«	Doppelbyte-POKE im Low/High-Format
»F Adresse, Anzahl, Byte«	Füllen eines Speicherbereiches
»T Adresse, "String"«	Speichert ab Adresse »Adresse« einen String im ASCII-Format ab

Suchfunktion

»Happy-Mon« ermöglicht es, den RAM-Speicher nach Einzel- oder Doppelbyte zu durchsuchen. Um ein einzelnes Byte zu finden, geben Sie folgenden Befehl ein: »HB Anfangsadresse, Byte (,N)«. Nun beginnt ab Adresse »Anfangsadresse« die Suche nach dem eingegebenen Byte, und gegebenenfalls wird die zugehörige Adresse im editierfähigen Format ausgegeben. Entsprechend verfährt der Befehl »HW Anfangsadresse, Word (,N)« beim Durchsuchen des RAM-Speichers nach einem Doppelbyte.

Ein- und Ausgabe

Alle auf dem Bildschirm ausgegebenen Texte lassen sich auch auf einem Drucker wiedergeben. Ein zweiter Ausgabekanal öffnet sich mit »O #Filespezifikation«, wobei Kanal Nummer 2 verwendet wird. Mit dem Befehl »N« kann man den Kanal wieder schließen. Als zweiten Ausgabekanal sollte man niemals den Editor oder den Bildschirm anwählen.

Ergänzen Sie jetzt Ihre HAPPY-COMPUTER-Sammlung

Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!



Kennen Sie alle »Happy Computer«-Ausgaben von 1985? Suchen Sie einen ganz bestimmten Testbericht? Oder haben Sie einen Teil eines interessanten Kurses versäumt? Suchen Sie nach einer speziellen Anwendung?
Damit Sie jetzt fehlende Hefte mit »Ihrem« Artikel nachbestellen können, finden Sie auf diesen Seiten eine Zusammenstellung aller wesentlichen Artikel der Ausgaben 01 bis 06 und 08 bis 12/85.
Und so kommen Sie schnell an die noch lieferbaren Ausgaben: Prüfen Sie, welche Ausgabe in Ihrer Sammlung noch fehlt, oder welches Thema Sie interessiert. Tragen Sie die Nummer dieser Ausgabe und das Erscheinungsjahr (z.B. 2/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird nach Zahlungseingang umgehend zur Auslieferung gebracht.

```
0000:FF FF 00 80 00 99 4C BF<71>
0008:83 48 68 C9 0A 90 06 18<0B>
0010:69 37 4C 12 80 18 69 30<9A>
0018:4C 17 82 A2 00 86 85 C9<28>
0020:30 90 0C C9 3A 90 0D C9<1D>
0028:41 90 04 C9 47 90 09 A0<28>
0030:01 84 85 60 38 E9 30 60<4D>
0038:38 E9 37 60 A0 00 84 86<07>
0040:B1 81 20 15 80 A6 85 E0<CB>
0048:00 D0 06 C8 E6 86 4C 3A<D5>
0050:80 A4 86 B1 81 C9 9B F0<43>
0058:13 C9 20 F0 0F C9 3A F0<17>
0060:0B C9 2C F0 07 68 68 A0<E7>
0068:01 4C 5D 82 60 A2 00 A1<D2>
0070:81 20 15 80 85 83 A5 85<57>
0078:D0 ED 20 30 83 60 20 67<D3>
0080:80 A5 83 0A 0A 0A 0A 85<4C>
0088:8B 20 67 80 A5 83 18 65<F3>
0090:8B 85 83 60 20 67 80 A5<4E>
0098:83 85 84 20 78 80 60 20<CA>
00A0:78 80 A5 83 85 84 20 78<BC>
00A8:80 60 A9 00 85 83 85 84<B3>
00B0:A2 00 A1 81 C9 2A D0 0C<68>
00B8:A5 97 85 83 A5 98 85 84<D2>
00C0:20 30 83 60 A1 81 C9 2B<EC>
00C8:D0 0B A5 9E 85 83 A5 9F<02>
00D0:85 84 4C BA 80 20 36 80<C8>
00D8:A5 86 C9 01 D0 03 4C 67<3F>
00E0:80 C9 02 D0 03 4C 78 80<C8>
00E8:C9 03 D0 03 4C 8E 80 C9<9A>
00F0:04 D0 03 4C 99 80 68 68<0B>
00F8:A0 01 4C 5D 82 20 A4 80<C4>
0100:A5 84 D0 01 60 68 68 A0<09>
0108:03 4C 5D 82 A2 00 A1 81<19>
0110:C9 2A D0 0C 20 30 83 A5<87>
0118:97 85 83 A5 98 85 84 60<01>
0120:A1 81 C9 2B D0 0C A5 9E<74>
0128:85 83 A5 9F 85 84 20 30<C2>
0130:83 60 A5 81 85 F3 A5 82<3D>
0138:85 F4 A9 00 85 F2 18 20<4B>
0140:00 D8 B0 2A 20 D2 D9 B0<AA>
0148:25 A5 D4 85 83 A5 D5 85<BA>
0150:84 A0 00 B1 81 C9 2E 90<5A>
0158:0A C9 39 90 02 D0 04 C8<D1>
0160:4C 4D 81 C8 98 18 65 80<4F>
0168:85 80 20 22 83 60 68 68<85>
0170:A0 02 4C 5D 82 20 06 81<33>
0178:A5 84 D0 01 60 68 68 A0<01>
0180:03 4C 5D 82 A2 00 A1 81<A2>
0188:C9 24 F0 03 4C 06 81 20<3A>
0190:30 83 4C A4 80 A2 00 A1<42>
0198:81 C9 24 F0 03 4C 6F 81<BA>
01A0:20 30 83 4C F7 80 A5 83<45>
01A8:18 4A 4A 4A 4A 20 03 80<47>
01B0:A5 83 29 0F 20 03 80 60<68>
01B8:A5 83 85 8C A5 84 85 83<1C>
01C0:20 A0 81 A5 8C 85 83 20<EC>
01C8:A0 81 AD 9A 86 C9 E1 F0<43>
01D0:05 A9 FF 8D 23 02 60 A5<ED>
01D8:83 85 D4 A5 84 85 D5 20<42>
01E0:AA D9 20 E6 D8 A0 00 B1<46>
01E8:F3 85 99 98 48 A5 99 29<65>
01F0:7F 20 17 82 68 A8 B1 F3<67>
01F8:30 04 C8 4C E1 81 60 A9<B8>
0200:0B 9D 42 03 A9 88 9D 44<52>
0208:03 A9 00 9D 45 03 9D 49<7A>
0210:03 A9 01 9D 48 03 A9 01<7C>
0218:85 11 4C 56 E4 85 88 A2<E8>
0220:00 20 F9 81 A5 87 F0 09<7F>
0228:A2 20 20 F9 81 C0 01 30<19>
0230:01 60 68 68 4C 5D 82 98<82>
0238:48 A9 20 20 17 82 68 A8<BA>
0240:88 C0 00 D0 F2 60 A0 00<87>
0248:B1 8B C9 9B F0 0F 85 88<66>
0250:98 48 A5 88 20 17 82 68<11>
0258:A8 C8 4C 42 82 60 A9 9B<45>
0260:4C 17 82 84 85 C0 80 90<06>
0268:13 A2 20 A9 0C 9D 42 03<39>
0270:20 56 E4 10 03 4C 2C 82<36>
0278:A9 00 85 87 A5 87 85 95<BF>
0280:A9 00 85 87 4C 89 82 FD<DC>
0288:45 52 52 4F 52 20 9B A9<A3>
0290:81 85 8B A9 82 85 8C 20<F1>
0298:40 82 A5 85 85 83 A9 00<93>
02A0:85 84 20 D1 81 4C A6 82<65>
02A8:20 28 24 9B A9 A2 85 8B<16>
02B0:A9 82 85 8C 20 40 82 20<38>
02B8:A0 81 A9 29 20 17 82 20<14>
02C0:58 82 A5 95 85 87 4C 6B<7F>
02C8:84 A9 00 85 8B 85 8C AA<74>
02D0:A5 8F C5 8B 08 A5 90 C5<D7>
02D8:8C 08 68 85 9D 68 25 9D<7B>
02E0:48 28 D0 05 A0 05 4C 5D<E1>
02E8:82 98 81 89 E6 89 D0 02<2F>
02F0:E6 8A 20 9C 8E E6 8B D0<FB>
02F8:02 E6 8C A5 8B C5 8F 08<C9>
0300:A5 8C C5 90 08 68 85 9D<25>
0308:68 25 9D 48 28 D0 DA 60<49>
0310:A2 80 A0 00 B1 81 C9 20<98>
0318:D0 0D 20 30 83 CA E0 00<47>
0320:D0 F2 AD 8B 86 F0 00 60<2C>
0328:18 A9 36 65 80 85 81 A9<35>
0330:85 69 00 85 82 60 E6 80<48>
0338:D0 02 E6 81 4C 22 83 20<8D>
0340:0A 83 A0 00 B1 81 C9 2C<71>
0348:D0 03 20 30 83 4C 0A 83<DA>
0350:A9 00 8D 44 02 D8 A9 19<6C>
0358:8D D9 02 A9 03 8D DA 02<A9>
0360:A9 B6 85 0C A9 85 85 0D<0F>
0368:A5 10 09 80 85 10 8D 0E<7B>
0370:D2 A5 93 8D C6 02 A5 94<D7>
0378:8D C5 02 A9 00 8D C8 02<C8>
0380:AD 8B 86 C9 B4 D0 0E AD<13>
0388:91 86 C9 80 D0 07 AD 9A<A8>
0390:86 C9 E1 F0 03 4C 71 E4<33>
0398:AD A9 86 C9 53 D0 F6 18<CA>
03A0:A5 58 69 50 8D AD 86 A9<46>
03A8:00 65 59 8D AE 86 A9 A5<49>
03B0:8D 30 02 A9 86 8D 31 02<DA>
03B8:A9 01 85 52 A9 27 85 53<50>
03C0:A9 00 85 4D 60 A9 00 85<9B>
03C8:97 85 98 85 9E 85 9F D8<CA>
03D0:AD A9 86 C9 53 F0 03 4C<40>
03D8:77 E4 A5 93 C9 02 F0 04<0B>
03E0:C9 08 D0 0A A5 94 C9 02<F2>
03E8:F0 0C C9 0A F0 08 A9 02<96>
03F0:85 93 A9 0A 85 94 A5 0D<66>
03F8:C9 85 F0 1A A5 0C 8D B4<46>
0400:84 A5 0D 8D B5 84 A2 00<25>
0408:B5 80 9D B6 84 E8 E0 81<DE>
0410:D0 F6 A9 FF 85 08 A2 00<52>
0418:A9 0C 9D 42 03 20 56 E4<D4>
0420:10 03 4C 2C 82 A9 80 85<73>
0428:6A A2 00 A9 03 9D 42 03<6E>
0430:A9 0C 9D 4A 03 A9 00 9D<1F>
0438:4B 03 4C 3A 84 45 3A 00<7D>
0440:A9 37 9D 44 03 A9 84 9D<B1>
0448:45 03 20 56 E4 10 03 4C<CE>
0450:2C 82 A2 20 A9 0C 9D 42<F7>
0458:03 20 56 E4 10 03 4C 2C<42>
0460:82 A9 00 85 87 A9 01 85<11>
0468:55 A9 00 85 56 A9 02 85<FB>
0470:54 20 4A 83 A9 FF 85 08<FF>
0478:A9 01 85 09 A9 01 85 55<5E>
0480:A9 00 8D 44 02 A9 22 8D<A5>
0488:2F 02 A5 87 85 95 A9 00<8F>
0490:85 87 20 58 82 4C 99 84<99>
0498:57 65 6C 6C 20 3F 9B A9<D6>
04A0:92 85 8B A9 84 85 8C 20<64>
04A8:40 82 A5 95 85 87 20 58<35>
04B0:82 A9 FF 8D FC 02 4C 28<85>
04B8:87 00 40 15 00 00 00 00<5D>
04C0:00 00 00 00 00 00 00 00<C8>
04C8:00 00 00 00 00 00 00 00<D0>
04D0:00 00 00 00 00 00 00 00<D8>
04D8:00 00 00 00 00 00 00 00<E0>
04E0:00 00 00 00 00 00 00 00<E8>
04E8:00 00 00 00 00 00 00 00<F0>
04F0:00 00 00 00 00 00 00 00<F8>
04F8:00 00 00 00 00 00 00 00<00>
0500:00 00 00 00 00 00 00 00<0A>
0508:00 00 00 00 00 00 00 00<12>
0510:00 00 00 00 00 00 00 00<1A>
0518:00 00 00 00 00 00 00 00<22>
0520:00 00 00 00 00 00 00 00<2A>
0528:00 00 00 00 00 00 00 00<32>
0530:00 00 00 00 00 00 00 00<3A>
0538:00 00 00 00 00 00 00 00<42>
0540:00 00 00 00 00 00 00 00<4A>
0548:00 00 00 00 00 00 00 00<52>
0550:00 00 00 00 00 00 00 00<5A>
0558:00 00 00 00 00 00 00 00<62>
0560:00 00 00 00 00 00 00 00<6A>
0568:00 00 00 00 00 00 00 00<72>
0570:00 00 00 00 00 00 00 00<7A>
0578:00 00 00 00 00 00 00 00<82>
0580:00 00 00 00 00 00 00 00<8A>
0588:00 00 00 00 00 00 00 00<92>
0590:00 00 00 00 00 00 00 00<9A>
0598:00 00 00 00 00 00 00 00<A2>
05A0:00 00 00 00 00 00 00 00<AA>
05A8:00 00 00 00 00 00 00 00<B2>
05B0:00 00 00 00 00 00 00 00<BA>
05B8:00 00 00 00 20 50 86 A9<B5>
05C0:00 8D B3 84 A9 28 8D D9<A6>
05C8:02 A9 05 8D DA 02 A9 92<63>
05D0:8D 36 02 A9 C0 8D 37 02<20>
05D8:A9 80 85 6A A2 20 A9 0C<39>
05E0:9D 42 03 20 56 E4 10 03<51>
05E8:4C 2C 82 A9 00 85 87 AD<5C>
05F0:1F D0 C9 03 F0 5D A2 00<BD>
05F8:BD B6 84 95 80 E8 E0 81<10>
0600:D0 F6 A9 02 85 52 A9 00<E3>
0608:8D 44 02 AD FB BF D0 03<6C>
0610:6C 0A 00 A2 00 A9 0C 9D<56>
0618:42 03 20 56 E4 10 03 4C<20>
0620:2C 82 A2 00 A9 03 9D 42<05>
0628:03 A9 0C 9D 4A 03 A9 00<CA>
0630:9D 4B 03 4C 33 86 45 3A<EA>
0638:00 A9 30 9D 44 03 A9 86<8C>
0640:9D 45 03 20 56 E4 10 03<AB>
0648:4C 2C 82 AD B3 84 D0 C0<6D>
0650:6C FA BF 4C BF 83 6C B4<B8>
0658:84 80 80 80 80 80 80 80<99>
0660:80 80 8D 8D 8D 80 A8 A1<D9>
0668:B0 B0 B9 8D AD AF AE 80<5D>
0670:92 8E 90 80 8D 8D 8D 80<2A>
0678:80 80 80 80 80 80 80 80<7B>
0680:80 80 80 80 80 80 88 E3<E5>
0688:89 91 99 98 95 80 E2 F9<C1>
0690:80 B4 E8 EF ED E1 F3 80<06>
0698:A6 E9 F3 E3 E8 E5 F2 ED<72>
06A0:E1 EE EE 80 80 80 80 80<D9>
06A8:80 80 80 70 70 70 42 53<E4>
06B0:86 02 42 00 00 02 02 02<96>
06B8:02 02 02 02 02 02 02 02<C4>
06C0:02 02 02 02 02 02 02 02<CC>
06C8:02 02 41 A5 86 A9 FF 85<26>
06D0:A0 4C 92 C0 A9 00 85 A0<EC>
06D8:A9 C7 8D 36 02 A9 86 8D<48>
06E0:37 02 A9 9B A2 00 9D 36<E5>
06E8:85 E8 E0 80 D0 F8 A9 00<D4>
06F0:8D 44 02 A2 00 A9 05 9D<08>
06F8:42 03 A9 36 9D 44 03 A9<1F>
0700:85 9D 45 03 A9 7F 9D 48<BB>
0708:03 A9 00 9D 49 03 20 56<86>
0710:E4 10 1A A9 01 85 55 C0<68>
0718:88 D0 05 A0 88 4C 5D 82<AB>
0720:20 4A 83 68 68 A9 FD 20<5F>
0728:17 82 4C 6B 84 60 20 CE<42>
0730:86 A9 00 85 80 20 22 83<FB>
0738:20 39 83 A2 00 A1 81 C9<8B>
0740:51 D0 08 A9 00 8D B3 84<92>
0748:4C BE 85 C9 2A D0 03 4C<0B>
0750:EC 8D C9 4C D0 07 A9 FF<82>
0758:85 9B 4C 51 8F C9 55 D0<FE>
0760:03 4C 43 98 C9 5A D0 07<F8>
0768:A9 00 85 9B 4C 51 8F C9<B0>
0770:41 D0 03 4C 0A 90 C9 45<1A>
0778:D0 03 4C CE 90 C9 44 D0<0C>
0780:03 4C 11 94 C9 9B D0 03<E5>
0788:4C 6B 84 C9 2B D0 03 4C<3E>
0790:D0 8E C9 4F D0 03 4C 4E<B1>
0798:88 C9 52 D0 03 4C B4 88<7C>
07A0:C9 57 D0 03 4C F9 88 C9<31>
07A8:4D D0 03 4C 3E 89 C9 50<68>
07B0:D0 03 4C DE 89 C9 56 D0<C2>
07B8:03 4C FE 89 C9 46 D0 03<A5>
07C0:4C 24 8A C9 54 D0 03 4C<EF>
07C8:54 8A C9 48 D0 03 4C 9C<97>
07D0:8A C9 3F D0 03 4C 00 8D<8D>
07D8:C9 4B D0 03 4C 29 8D C9<31>
07E0:58 D0 08 A9 01 8D B3 84<3F>
07E8:4C BE 85 C9 47 D0 03 4C<14>
07F0:45 8D C9 4E D0 03 4C D6<4B>
07F8:8D C9 3E F0 05 A0 06 4C<E4>
0800:5D 82 20 30 83 20 39 83<90>
0808:A2 00 A1 81 20 15 80 A5<A0>
0810:85 F0 0C A2 00 A1 81 C9<B8>
0818:2A F0 04 C9 40 D0 0E 20<BF>
0820:A4 80 20 39 83 A2 00 A1<FD>
0828:81 C9 3A F0 05 A0 07 4C<72>
0830:5D 82 20 30 83 20 39 83<E0>
0838:A2 00 A1 81 20 15 80 A5<30>
0840:85 D0 EA 20 30 83 A2 00<94>
0848:A1 81 20 15 80 A5 85 D0<87>
0850:DC 4C 4B 8F 20 30 83 20<9A>
0858:39 83 A2 00 A1 81 C9 23<7D>
0860:F0 05 A0 09 4C 5D 82 20<43>
0868:30 83 A2 20 A9 0C 9D 42<19>
0870:03 20 56 E4 10 03 4C 2C<62>
0878:82 A9 03 A2 20 9D 42 03<B9>
0880:A9 00 9D 4B 03 A5 81 9D<18>
0888:44 03 A5 82 9D 45 03 A9<E9>
0890:08 9D 4A 03 20 56 E4 10<77>
0898:1A 98 48 A2 20 A9 0C 9D<07>
08A0:42 03 20 56 E4 10 03 4C<AD>
08A8:2C 82 A9 00 85 87 68 A8<3F>
08B0:4C 2C 82 A9 01 85 87 4C<35>
08B8:6B 84 20 30 83 20 39 83<D0>
08C0:20 7E 81 A5 83 8D 0A 03<30>
08C8:A5 84 8D 0B 03 20 39 83<FC>
```

**Listing zu »Happy-Mon«.
Bitte mit AMPEL eingeben.**

```
08D0:20 7E 81 A5 83 8D 04 03<14>
08D8:A5 84 8D 05 03 A5 83 85<34>
08E0:89 A5 84 85 8A 20 9C 8E<A9>
08E8:E6 8A 20 9C 8E A9 52 8D<60>
08F0:02 03 20 53 E4 C0 00 10<29>
08F8:03 4C 2C 82 4C 6B 84 20<63>
0900:30 83 20 39 83 20 7E 81<6C>
0908:A5 83 8D 0A 03 A5 84 8D<20>
0910:0B 03 20 39 83 20 7E 81<12>
0918:A5 83 8D 04 03 A5 84 8D<B0>
0920:05 03 A5 83 85 89 A5 84<4F>
0928:85 8A 20 9C 8E E6 8A 20<44>
0930:9C 8E A9 57 8D 02 03 20<D0>
0938:53 E4 C0 00 10 03 4C 2C<34>
0940:82 4C 6B 84 20 30 83 20<C2>
0948:39 83 20 7E 81 A5 83 85<9E>
0950:8D A5 84 85 8E A5 83 85<2B>
0958:89 A5 84 85 8A 20 9C 8E<2A>
0960:20 39 83 20 7E 81 A5 83<66>
0968:85 89 A5 84 85 8A 20 9C<EA>
0970:8E 20 39 83 20 7E 81 A5<C0>
0978:83 85 8F A5 84 85 90 A9<EA>
0980:00 85 8B 85 8C AA A5 8B<76>
0988:C5 8F 08 A5 8C C5 90 08<17>
0990:68 85 9D 68 25 9D 48 28<94>
0998:D0 03 4C DE 82 A1 8D 81<EB>
09A0:89 E6 8D D0 02 E6 8E A5<F7>
09A8:89 48 A5 8A 48 A5 8D 85<FD>
09B0:89 A5 8E 85 8A 20 9C 8E<14>
09B8:68 85 8A 68 85 89 E6 89<97>
09C0:D0 02 E6 8A 20 9C 8E E6<03>
09C8:8B D0 02 E6 8C A5 8B C5<83>
09D0:8F 08 A5 8C C5 90 08 68<6C>
09D8:85 9D 68 25 9D 48 28 D0<B9>
09E0:BC 4C 6B 84 20 30 83 20<80>
09E8:39 83 20 7E 81 A5 83 85<FE>
09F0:89 A5 84 85 8A 20 39 83<C1>
09F8:20 8F 81 A2 00 A5 83 81<46>
0A00:89 4C 6B 84 20 30 83 20<78>
0A08:39 83 20 7E 81 A5 83 85<59>
0A10:89 A5 84 85 8A 20 39 83<A3>
0A18:20 7E 81 A0 00 A5 83 91<D0>
0A20:89 A0 01 A5 84 91 89 4C<1E>
0A28:6B 84 20 30 83 20 39 83<6B>
0A30:20 7E 81 A5 83 85 89 A5<3C>
0A38:84 85 8A 20 9C 8E 20 39<36>
0A40:83 20 7E 81 A5 83 85 8F<1E>
0A48:A5 84 85 90 20 39 83 20<34>
0A50:8F 81 A4 83 20 C3 82 4C<D5>
0A58:6B 84 20 30 83 20 39 83<1B>
0A60:20 7E 81 A5 83 85 89 A5<0C>
0A68:84 85 8A 20 9C 8E 20 39<26>
0A70:83 A2 00 A1 81 C9 22 F0<5C>
0A78:03 4C 5C 88 20 30 83 A1<8E>
0A80:81 C9 22 F0 12 C9 9B F0<A5>
0A88:0E 81 89 E6 89 D0 02 E6<C8>
0A90:8A 20 9C 8E 4C 76 8A A5<3C>
0A98:89 85 97 A5 8A 85 98 4C<9F>
0AA0:6B 84 A9 FF 85 9C 20 30<5F>
0AA8:83 20 39 83 A2 00 A1 81<66>
0AB0:C9 42 F0 0A C9 57 D0 03<A4>
0AB8:4C 48 8C 4C 5C 88 20 30<8C>
0AC0:83 20 39 83 20 7E 81 A5<FC>
0AC8:83 85 89 A5 84 85 8A A5<D9>
0AD0:83 85 9A A5 84 85 9B 20<6C>
0AD8:9C 8E 20 39 83 20 8F 81<CA>
0AE0:A5 83 85 91 20 39 83 A2<6E>
0AE8:00 A1 81 C9 4E D0 04 A9<48>
0AF0:00 85 9C 20 4B 8E A2 00<D2>
0AF8:A1 89 C5 91 D0 2B A9 3E<A1>
0B00:20 17 82 A5 89 85 83 A5<88>
0B08:8A 85 84 20 B2 81 A9 3A<37>
0B10:20 17 82 A5 91 85 83 20<32>
0B18:A0 81 20 58 82 A5 9C F0<DB>
0B20:08 20 6D 8B F0 03 4C 6B<ED>
0B28:84 E6 89 D0 02 E6 8A 20<A2>
0B30:AB 8E A5 89 C5 9A 08 A5<AE>
0B38:8A C5 9B 08 68 85 9D 68<2E>
0B40:25 9D 48 28 D0 AD 20 58<65>
0B48:82 4C 5C 8B 57 68 6F 6C<05>
0B50:65 20 6D 65 6D 6F 72 79<15>
0B58:20 63 68 65 63 6B 65 64<51>
0B60:2E 9B A9 46 85 8B A9 8B<9C>
0B68:85 8C 20 40 82 20 58 82<AF>
0B70:4C 6B 84 A5 87 85 95 A9<44>
0B78:00 85 87 4C 8C 8B 9C 9D<E2>
0B80:43 6F 6E 74 69 6E 75 65<1B>
0B88:20 28 59 2F 4E 29 3F 59<9B>
0B90:1E 9B A9 78 85 8B A9 8B<2F>
0B98:85 8C 20 40 82 A9 FF 8D<9A>
0BA0:FC 02 A2 10 A9 0C 9D 42<63>
0BA8:03 20 56 E4 10 03 4C 2C<A0>
0BB0:82 AD 30 02 C9 A5 F0 07<E0>
0BB8:A9 FF 85 09 4C B6 8B A2<14>
0BC0:10 A9 03 9D 42 03 A9 04<A6>
0BC8:9D 4A 03 A9 00 9D 4B 03<A6>
0BD0:4C D0 8B 4B 3A 00 A9 CD<90>
0BD8:9D 44 03 A9 8B 9D 45 03<57>
0BE0:20 56 E4 10 03 4C 2C 82<BD>
0BE8:A9 07 9D 42 03 A9 96 9D<53>
0BF0:44 03 A9 00 9D 45 03 9D<B7>
0BF8:49 03 A9 01 9D 48 03 20<CC>
0C00:56 E4 10 18 A9 9C 20 17<0E>
0C08:82 A9 FD 20 17 82 C0 88<F7>
0C10:D0 03 4C 75 8B A5 95 85<40>
0C18:87 4C 6B 84 A2 10 A9 0C<27>
0C20:9D 42 03 20 56 E4 10 03<E2>
0C28:4C 2C 82 A9 9C 20 17 82<49>
0C30:A5 95 85 87 A5 96 C9 4E<6C>
0C38:D0 03 C9 00 60 C9 59 F0<DA>
0C40:0C C9 9B F0 08 A9 FD 20<DF>
0C48:17 82 4C 0C 8C 60 20 30<E0>
0C50:83 20 39 83 20 7E 81 A5<A1>
0C58:83 85 89 A5 84 85 8A A5<26>
0C60:83 85 9A A5 84 85 9B 20<E9>
0C68:9C 8E 20 39 83 20 7E 81<24>
0C70:A5 83 85 91 A5 84 85 92<A4>
0C78:20 39 83 A2 00 A1 81 C9<CA>
0C80:4E D0 04 A9 00 85 9C 20<38>
0C88:4B 8E A5 10 09 80 85 10<FE>
0C90:8D 0E D2 20 AB 8E A2 00<02>
0C98:A1 89 C5 91 D0 45 E6 89<09>
0CA0:D0 02 E6 8A A2 00 A1 89<6D>
0CA8:C5 92 D0 3D A9 3E 20 17<EA>
0CB0:82 A5 89 85 83 A5 8A 85<9F>
0CB8:84 20 B2 81 A9 3A 20 17<76>
0CC0:82 A5 91 85 83 20 A0 81<A2>
0CC8:A0 01 20 31 82 A5 92 85<5C>
0CD0:83 20 A0 81 20 58 82 A5<46>
0CD8:9C F0 0E 20 6D 8B F0 09<F6>
0CE0:4C 6B 84 E6 89 D0 02 E6<C1>
0CE8:8A 20 AB 8E A5 89 C5 9A<FA>
0CF0:08 A5 8A C5 9B 08 68 85<43>
0CF8:9D 68 25 9D 48 28 D0 03<E6>
0D00:4C 40 8B 4C 81 8C 20 30<08>
0D08:83 20 39 83 20 7E 81 A9<F2>
0D10:24 20 17 82 20 B2 81 A0<75>
0D18:01 20 31 82 A9 3D 20 17<82>
0D20:82 A0 01 20 31 82 20 D1<26>
0D28:81 20 58 82 4C 6B 84 A5<71>
0D30:94 C9 0A D0 0B A9 02 85<0C>
0D38:94 A9 08 85 93 4C 6B 84<BE>
0D40:A9 0A 85 94 A9 02 85 93<F6>
0D48:4C 6B 84 20 30 83 20 39<7A>
0D50:83 A2 00 A1 81 C9 2A F0<99>
0D58:0B 20 7E 81 A5 83 85 97<8C>
0D60:A5 84 85 98 A5 87 85 95<01>
0D68:A9 00 85 87 20 58 82 20<D7>
0D70:6D 8B F0 21 4C 7E 8D 4E<F5>
0D78:6F 74 20 65 78 65 63 75<46>
0D80:74 65 64 9B A9 71 85 8B<03>
0D88:A9 8D 85 8C 20 40 82 20<08>
0D90:58 82 4C 6B 84 4C A1 8D<AF>
0D98:52 75 6E 6E 69 6E 67 20<B9>
0DA0:66 72 6F 6D 20 24 9B A9<78>
0DA8:92 85 8B A9 8D 85 8C 20<A2>
0DB0:40 82 A5 97 85 83 A5 98<84>
0DB8:85 84 20 B2 81 4C BF 8D<95>
0DC0:20 2E 2E 2E 9B A9 BA 85<8F>
0DC8:8B A9 8D 85 8C 20 40 82<1B>
0DD0:20 58 82 20 D3 8D 4C 6B<D1>
0DD8:84 6C 97 00 A2 20 A9 0C<55>
0DE0:9D 42 03 20 56 E4 10 03<A1>
0DE8:4C 2C 82 A9 00 85 87 4C<6B>
0DF0:6B 84 20 30 83 20 39 83<2D>
0DF8:20 7E 81 A5 83 85 97 A5<0A>
0E00:84 85 98 4C 6B 84 0F D4<CA>
0E08:00 D8 A9 00 85 85 AD 31<EF>
0E10:02 C9 86 F0 07 A9 00 85<7E>
0E18:81 6C 0A 00 A5 89 CD 00<FB>
0E20:8E 08 A5 8A CD 01 8E 08<E1>
0E28:68 85 9D 68 25 9D 48 28<FB>
0E30:90 02 D0 01 60 A5 89 CD<0C>
0E38:02 8E 08 A5 8A CD 03 8E<5B>
0E40:08 68 85 9D 68 25 9D 48<27>
0E48:28 90 01 60 A9 04 85 85<5D>
0E50:60 20 04 8E A5 85 F0 49<ED>
0E58:20 58 82 A9 01 85 55 A9<46>
0E60:01 85 52 4C 85 8E 2D 2D<F2>
0E68:3E 20 24 44 34 31 30 20<5B>
0E70:2D 20 24 44 37 46 46 20<63>
0E78:63 61 6E 6E 6F 74 20 62<0E>
0E80:65 20 61 63 63 65 73 73<A4>
0E88:65 64 9B A9 60 85 8B A9<2D>
0E90:8E 85 8C 20 40 82 20 58<28>
0E98:82 A9 00 85 89 A9 D8 85<A0>
0EA0:8A 60 20 04 8E A5 85 F0<D7>
0EA8:07 68 68 A0 04 4C 5D 82<F7>
0EB0:60 A5 A0 F0 20 68 68 A5<4F>
0EB8:87 85 95 A9 00 85 87 A9<60>
0EC0:FD 20 17 82 A5 95 85 87<75>
0EC8:AD AA 86 C9 86 F0 03 4C<48>
0ED0:77 E4 4C 6B 84 60 20 30<67>
0ED8:83 20 39 83 20 7E 81 A5<1C>
0EE0:83 85 9E A5 84 85 9F 4C<28>
0EE8:6B 84 A9 00 85 80 20 22<32>
0EF0:83 20 39 83 20 30 83 20<0D>
0EF8:39 83 20 A4 80 A5 83 85<BA>
0F00:89 A5 84 85 8A 20 9C 8E<76>
0F08:A5 89 85 97 A5 8A 85 98<03>
0F10:20 39 83 20 30 83 A0 00<87>
0F18:98 48 20 39 83 A2 00 A1<EF>
0F20:81 20 15 80 A5 85 D0 11<51>
0F28:20 78 80 68 A8 A5 83 91<56>
0F30:97 C8 C0 08 F0 05 4C 12<40>
0F38:8F 68 A8 84 8F 18 A5 97<9A>
0F40:65 8F 85 97 A9 00 65 98<24>
0F48:85 98 20 A5 90 20 4A 83<FF>
0F50:60 20 E4 8E 4C 28 87 20<BD>
0F58:30 83 20 39 83 20 7E 81<97>
0F60:A5 83 85 89 A5 84 85 8A<A7>
0F68:20 9C 8E A5 89 85 97 A5<EC>
0F70:8A 85 98 A9 FF 85 9C 20<15>
0F78:39 83 A2 00 A1 81 C9 4E<FA>
0F80:D0 04 A9 00 85 9C 20 86<19>
0F88:8F 4C 80 8F A5 97 85 89<63>
0F90:A5 98 85 8A 20 4B 8E A5<60>
0F98:89 85 97 A5 8A 85 98 A9<FA>
0FA0:3E 20 17 82 A5 97 85 83<F6>
0FA8:A5 98 85 84 20 B2 81 A9<7D>
0FB0:3A 20 17 82 A0 00 B1 97<F4>
0FB8:85 83 98 48 20 A0 81 A0<83>
0FC0:01 20 31 82 68 A8 C8 C0<D3>
0FC8:08 D0 EB A5 9B F0 21 A0<FD>
0FD0:00 B1 97 85 9A 98 48 A9<9A>
0FD8:1B 20 17 82 A5 9A C9 9B<C3>
0FE0:D0 04 A9 20 85 88 20 17<B8>
0FE8:82 68 A8 C8 C0 08 D0 E1<91>
0FF0:A2 00 E6 97 D0 02 E6 98<71>
0FF8:E8 E0 08 D0 F5 20 58 82<5E>
1000:20 AB 8E A5 9C F0 05 20<9D>
1008:6D 8B D0 01 60 4C 6B 84<38>
1010:20 30 83 20 39 83 20 7E<4C>
1018:81 A5 83 85 89 A5 84 85<55>
1020:8A 20 9C 8E A5 89 85 97<5A>
1028:A5 8A 85 98 A9 07 85 52<05>
1030:20 58 82 A9 07 85 52 A9<CB>
1038:01 85 55 A5 87 85 95 A9<37>
1040:00 85 87 A9 3E 20 17 82<2F>
1048:A5 97 85 83 A5 98 85 84<63>
1050:20 B2 81 4C 54 90 3A 1F<09>
1058:1E 9B A9 50 85 8B A9 90<45>
1060:85 8C 20 40 82 A5 95 85<BB>
1068:87 20 CE 86 A9 01 85 52<0A>
1070:A9 01 85 55 A9 00 85 80<7B>
1078:20 22 83 A2 00 A1 81 C9<FA>
1080:9B D0 15 4C 84 90 FD 1C<42>
1088:9C 9B A9 80 85 8B A9 90<F2>
1090:85 8C 20 40 82 4C 2D 90<85>
1098:20 39 83 20 78 80 A9 00<80>
10A0:85 80 20 22 83 20 10 8F<93>
10A8:4C 2D 90 A9 00 8D 08 D2<CA>
10B0:A9 03 8D 0F D2 A9 78 8D<3D>
10B8:00 D2 A9 A5 8D 01 D2 A9<71>
10C0:03 8D 20 02 AD 20 02 D0<6B>
10C8:FB A9 00 8D 00 D2 A9 00<02>
10D0:8D 01 D2 60 20 30 83 20<96>
10D8:39 83 20 7E 81 A5 83 85<FC>
10E0:89 A5 84 85 8A 20 9C 8E<99>
10E8:A5 89 85 97 A5 8A 85 98<29>
10F0:A9 07 85 52 20 58 82 A9<8C>
10F8:07 85 52 A9 01 85 55 A5<1D>
1100:87 85 95 A9 00 85 87 A9<12>
1108:FF 85 9B A9 00 85 9C 20<36>
1110:86 8F 38 A5 97 E9 08 85<4F>
1118:97 A5 98 E9 00 85 98 A9<60>
1120:07 85 55 A9 1C 20 17 82<7D>
1128:A5 95 85 87 20 CE 86 A9<D7>
1130:01 85 52 A9 01 85 55 A9<7A>
1138:00 85 80 20 22 83 20 39<42>
1140:83 20 78 80 A9 00 85 80<89>
1148:20 22 83 20 10 8F 4C F1<7B>
1150:90 42 52 4B 30 4F 52 41<29>
1158:37 3F 3F 3F 4F 52 41 33<A6>
1160:41 53 4C 33 3F 50 48 50<AE>
1168:30 4F 52 41 34 41 53 4C<4C>
1170:31 3F 3F 4F 52 41 32 41<EC>
1178:53 4C 32 3F 42 50 4C 42<AB>
1180:4F 52 41 38 3F 3F 3F 4F<2E>
1188:52 41 39 41 53 4C 39 3F<D4>
1190:43 4C 43 30 4F 52 41 36<3F>
```

Listing zu »Happy-Mon« (Fortsetzung)

```
1198:3F 3F 3F 4F 52 41 35 41<42>
11A0:53 4C 35 3F 4A 53 52 32<6E>
11A8:41 4E 44 37 3F 3F 42 49<9D>
11B0:54 33 41 4E 44 33 52 4F<6F>
11B8:4C 33 3F 50 4C 50 30 41<7D>
11C0:4E 44 34 52 4F 4C 31 3F<69>
11C8:42 49 54 32 41 4E 44 32<69>
11D0:52 4F 4C 32 3F 42 4D 49<02>
11D8:42 41 4E 44 38 3F 3F 3F<72>
11E0:41 4E 44 39 52 4F 4C 39<E3>
11E8:3F 53 45 43 30 41 4E 44<A0>
11F0:36 3F 3F 3F 41 4E 44 35<B7>
11F8:52 4F 4C 35 3F 52 54 49<08>
1200:30 45 4F 52 37 3F 3F 3F<D0>
1208:45 4F 52 33 4C 53 52 33<95>
1210:3F 50 48 41 30 45 4F 52<8A>
1218:34 4C 53 52 31 3F 4A 4D<5F>
1220:50 32 45 4F 52 32 4C 53<B3>
1228:52 32 3F 42 56 43 42 45<4D>
1230:4F 52 38 3F 3F 3F 45 4F<1B>
1238:52 39 4C 53 52 39 3F 43<3A>
1240:4C 49 30 45 4F 52 36 3F<FD>
1248:3F 3F 45 4F 52 35 4C 53<27>
1250:52 35 3F 52 54 53 30 41<00>
1258:44 43 37 3F 3F 3F 41 44<83>
1260:43 33 52 4F 52 33 3F 50<4F>
1268:4C 41 30 41 44 43 34 52<C2>
1270:4F 52 31 3F 4A 4D 50 43<15>
1278:41 44 43 32 52 4F 52 32<3E>
1280:3F 42 56 53 42 41 44 43<A5>
1288:38 3F 3F 3F 41 44 43 39<2E>
1290:52 4F 52 39 3F 53 45 49<A2>
1298:30 41 44 43 36 3F 3F 3F<39>
12A0:41 44 43 35 52 4F 52 35<C9>
12A8:3F 3F 53 54 41 37 3F 3F<70>
12B0:53 54 59 33 53 54 41 33<41>
12B8:53 54 58 33 3F 44 45 59<23>
12C0:30 3F 54 58 41 30 3F 53<38>
12C8:54 59 32 53 54 41 32 53<C3>
12D0:54 58 32 3F 42 43 43 42<CE>
12D8:53 54 41 38 3F 3F 53 54<AE>
12E0:59 39 53 54 41 39 53 54<6E>
12E8:58 41 3F 54 59 41 30 53<F0>
12F0:54 41 36 54 58 53 30 3F<09>
12F8:3F 53 54 41 35 3F 3F 4C<56>
1300:44 59 34 4C 44 41 37 4C<A7>
1308:44 58 34 3F 4C 44 59 33<25>
1310:4C 44 41 33 4C 44 58 33<1B>
1318:3F 54 41 59 30 4C 44 41<1C>
1320:34 54 41 58 30 3F 4C 44<6C>
1328:59 32 4C 44 41 32 4C 44<C2>
1330:58 32 3F 42 43 53 42 4C<D2>
1338:44 41 38 3F 3F 4C 44 59<57>
1340:39 4C 44 41 39 4C 44 58<F9>
1348:41 3F 43 4C 56 30 4C 44<BB>
1350:41 36 54 53 58 30 3F 4C<21>
1358:44 59 35 4C 44 41 35 4C<AB>
1360:44 58 35 3F 43 50 58 34<75>
1368:43 4D 50 37 3F 3F 43 50<7C>
1370:59 33 43 4D 50 33 44 45<66>
1378:43 33 3F 49 4E 59 30 43<00>
1380:4D 50 34 44 45 58 30 3F<55>
1388:43 50 59 32 43 4D 50 32<C2>
1390:44 45 43 32 3F 42 4E 45<96>
1398:42 43 4D 50 38 3F 3F 3F<D3>
13A0:43 4D 50 39 44 45 43 39<2B>
13A8:3F 43 4C 44 30 43 4D 50<64>
13B0:36 3F 3F 3F 43 4D 50 35<7E>
13B8:44 45 43 35 3F 43 50 58<89>
13C0:34 53 42 43 37 3F 3F 43<35>
13C8:50 58 33 53 42 43 33 49<16>
13D0:4E 43 33 3F 49 4E 58 30<2B>
13D8:53 42 43 34 4E 4F 50 30<C2>
13E0:3F 43 50 58 32 53 42 43<88>
13E8:32 49 4E 43 32 3F 42 45<A5>
13F0:51 42 53 42 43 38 3F 3F<52>
13F8:3F 53 42 43 39 49 4E 43<93>
1400:39 3F 53 45 44 30 53 42<0E>
1408:43 36 3F 3F 3F 53 42 43<29>
1410:35 49 4E 43 35 3F 3F 20<56>
1418:30 83 20 39 83 20 7E 81<D9>
1420:A5 83 85 89 A5 84 85 8A<5E>
1428:20 9C 8E A5 89 85 97 A5<93>
1430:8A 85 98 20 39 83 A9 00<9C>
1438:85 9C A9 00 85 A3 A2 00<B0>
1440:A1 81 C9 4E D0 04 A9 FF<B5>
1448:85 9C A5 97 85 89 A5 98<BF>
1450:85 8A 20 4B 8E A9 3E 20<61>
1458:17 82 A5 97 85 83 A5 98<D2>
1460:85 84 20 B2 81 A9 3A 20<DE>
1468:17 82 20 79 94 20 58 82<8E>
1470:20 AB 8E A5 9C D0 D3 20<45>
1478:6D 8B F0 CE 4C 6B 84 A2<CA>
1480:00 A1 97 85 83 A5 A3 D0<BD>
1488:03 20 A0 81 A2 00 A1 97<1B>
1490:85 9B E6 97 D0 02 E6 98<5A>
1498:A9 4B 85 91 A9 91 85 92<CC>
14A0:A0 00 C4 9B F0 23 A2 00<1A>
14A8:A1 91 C9 3F D0 0A E6 91<65>
14B0:D0 02 E6 92 C8 4C 9C 94<73>
14B8:18 A5 91 69 04 85 91 A5<B7>
14C0:92 69 00 85 92 C8 4C 9C<F6>
14C8:94 A2 00 A1 91 C9 3F D0<33>
14D0:44 A5 A3 D0 1B 4C DE 94<8E>
14D8:20 20 20 20 20 20 2D 20<19>
14E0:3F 3F 3F 9B A9 D2 85 8B<31>
14E8:A9 94 85 8C 20 40 82 60<98>
14F0:4C FC 94 20 20 20 20 20<83>
14F8:20 20 20 2E 42 59 54 45<7C>
1500:20 9B A9 ED 85 8B A9 94<1C>
1508:85 8C 20 40 82 A9 00 85<13>
1510:84 20 D1 81 60 A0 03 B1<B7>
1518:91 20 15 80 C9 00 D0 03<26>
1520:4C 95 95 C9 01 D0 03 4C<50>
1528:A2 95 C9 02 D0 03 4C B4<58>
1530:95 C9 03 D0 03 4C F5 95<FC>
1538:C9 04 D0 03 4C 19 96 C9<AE>
1540:05 D0 03 4C 42 96 C9 06<03>
1548:D0 03 4C 8D 96 C9 07 D0<0B>
1550:03 4C D8 96 C9 08 D0 03<2C>
1558:4C 13 97 C9 09 D0 03 4C<B6>
1560:51 97 C9 0A D0 03 4C 7F<34>
1568:97 C9 0B D0 03 4C AD 97<36>
1570:4C FD 97 A5 A3 D0 05 A9<54>
1578:2D 20 17 82 A9 20 20 17<27>
1580:82 A0 00 B1 91 85 88 98<2C>
1588:48 A5 88 20 17 82 68 A8<48>
1590:C8 C0 03 D0 EE A0 01 20<FB>
1598:31 82 60 A5 A3 D0 05 A0<38>
15A0:06 20 31 82 20 6D 95 60<55>
15A8:A5 A3 D0 05 A0 06 20 31<A6>
15B0:82 20 6D 95 A9 41 20 17<50>
15B8:82 60 A2 00 A1 97 85 8D<E9>
15C0:E6 97 D0 02 E6 98 A1 97<27>
15C8:85 8E E6 97 D0 02 E6 98<B9>
15D0:A5 A3 D0 13 A5 8D 85 83<4E>
15D8:20 A0 81 A5 8E 85 83 20<3C>
15E0:A0 81 A0 02 20 31 82 20<B5>
15E8:6D 95 A9 24 20 17 82 A5<8C>
15F0:8D 85 83 A5 8E 85 84 20<76>
15F8:B2 81 60 A2 00 A1 97 85<D8>
1600:83 E6 97 D0 02 E6 98 A5<D3>
1608:A3 D0 08 20 A0 81 A0 04<02>
1610:20 31 82 20 6D 95 A9 24<FF>
1618:20 17 82 20 A0 81 60 A2<B0>
1620:00 A1 97 85 83 E6 97 D0<88>
1628:02 E6 98 A5 A3 D0 08 20<A0>
1630:A0 81 A0 04 20 31 82 20<6B>
1638:6D 95 A9 23 20 17 82 A9<4A>
1640:24 20 17 82 20 A0 81 60<23>
1648:A2 00 A1 97 85 8D E6 97<A4>
1650:D0 02 E6 98 A1 97 85 8E<27>
1658:E6 97 D0 02 E6 98 A5 A3<31>
1660:D0 13 A5 8D 85 83 20 A0<25>
1668:81 A5 8E 85 83 20 A0 81<25>
1670:A0 02 20 31 82 20 6D 95<80>
1678:A9 24 20 17 82 A5 8D 85<F6>
1680:83 A5 8E 85 84 20 B2 81<1C>
1688:A9 2C 20 17 82 A9 58 20<10>
1690:17 82 60 A2 00 A1 97 85<53>
1698:8D E6 97 D0 02 E6 98 A1<EE>
16A0:97 85 8E E6 97 D0 02 E6<6F>
16A8:98 A5 A3 D0 13 A5 8D 85<06>
16B0:83 20 A0 81 A5 8E 85 83<31>
16B8:20 A0 81 A0 02 20 31 82<5B>
16C0:20 6D 95 A9 24 20 17 82<87>
16C8:A5 8D 85 83 A5 8E 85 84<71>
16D0:20 B2 81 A9 2C 20 17 82<65>
16D8:A9 59 20 17 82 60 A2 00<77>
16E0:A1 97 85 83 E6 97 D0 02<96>
16E8:E6 98 A5 A3 D0 08 20 A0<7A>
16F0:81 A0 04 20 31 82 20 6D<80>
16F8:95 A9 28 20 17 82 A9 24<48>
1700:20 17 82 20 A0 81 4C 07<B4>
1708:97 2C 58 29 9B A9 03 85<34>
1710:8B A9 97 85 8C 20 40 82<FE>
1718:60 A2 00 A1 97 85 83 20<E0>
1720:39 83 E6 97 D0 02 E6 98<9C>
1728:A5 A3 D0 08 20 A0 81 A0<F0>
1730:04 20 31 82 20 6D 95 A9<2B>
1738:28 20 17 82 A9 24 20 17<7A>
1740:82 20 A0 81 4C 45 97 29<02>
1748:2C 59 9B A9 41 85 8B A9<3C>
1750:97 85 8C 20 40 82 60 A2<25>
1758:00 A1 97 85 83 E6 97 D0<CA>
1760:02 E6 98 A5 A3 D0 08 20<EA>
1768:A0 81 A0 04 20 31 82 20<44>
1770:6D 95 A9 24 20 17 82 20<82>
1778:A0 81 A9 2C 20 17 82 A9<F6>
1780:58 20 17 82 60 A2 00 A1<14>
1788:97 85 83 E6 97 D0 02 E6<18>
1790:98 20 A0 81 A5 A3 D0 05<F4>
1798:A0 04 20 31 82 20 6D 95<A9>
17A0:A9 24 20 17 82 20 A0 81<F8>
17A8:A9 2C 20 17 82 A9 59 20<35>
17B0:17 82 60 A2 00 A1 97 85<29>
17B8:83 85 9A E6 97 D0 02 E6<E0>
17C0:98 A5 A3 D0 08 20 A0 81<D8>
17C8:A0 04 20 31 82 20 6D 95<D9>
17D0:A9 24 20 17 82 A5 9A C9<56>
17D8:80 90 17 38 A9 00 E5 9A<73>
17E0:85 9A 38 A5 97 E5 9A 85<D1>
17E8:83 A5 98 E9 00 85 84 4C<09>
17F0:F9 97 18 A5 97 65 9A 85<A0>
17F8:83 A5 98 69 00 85 84 20<D5>
1800:B2 81 60 A2 00 A1 97 85<D6>
1808:8D E6 97 D0 02 E6 98 A1<C4>
1810:97 85 8E E6 97 D0 02 E6<FB>
1818:98 A5 A3 D0 13 A5 8D 85<AA>
1820:83 20 A0 81 A5 8E 85 83<7E>
1828:20 A0 81 A0 02 20 31 82<DF>
1830:20 6D 95 A9 28 20 17 82<3A>
1838:A5 8D 85 83 A5 8E 85 84<7E>
1840:20 B2 81 A9 29 20 17 82<D9>
1848:60 20 30 83 20 39 83 20<77>
1850:7E 81 A5 83 85 89 A5 84<C4>
1858:85 8A 20 9C 8E A5 89 85<56>
1860:97 A5 8A 85 98 4C 6F 98<8F>
1868:31 30 30 30 20 20 2E 4F<51>
1870:52 47 20 24 9B A9 62 85<99>
1878:8B A9 98 85 8C 20 40 82<B3>
1880:A5 97 85 83 A5 98 85 84<DA>
1888:20 B2 81 20 58 82 4C 92<72>
1890:98 31 30 30 31 20 3B 9B<EB>
1898:A9 BB 85 8B A9 98 85 8C<8B>
18A0:20 40 82 20 58 82 A9 00<41>
18A8:85 9C A9 FF 85 A3 A9 E9<10>
18B0:85 A1 A9 03 85 A2 20 39<C0>
18B8:83 A2 00 A1 81 C9 4E D0<42>
18C0:04 A9 FF 85 9C E6 A1 D0<CA>
18C8:02 E6 A2 A5 97 85 89 A5<78>
18D0:98 85 8A 20 4B 8E A5 A1<6B>
18D8:85 83 A5 A2 85 84 20 D1<B5>
18E0:81 A0 02 20 31 82 20 79<88>
18E8:94 20 58 82 20 AB 8E A5<09>
18F0:9C D0 D2 20 6D 8B F0 CD<F9>
18F8:4C 6B 84 00 00 00 00 00<B9>
1900:00 00 00 00 00 00 00 00<32>
Laenge 6407 Bytes
```

Listing zu »Happy-Mon« (Schluß)

»Happy-Mon« greift sogar auf einzelne Diskettensektoren zu. Dies ist speziell bei Boot-Disketten nützlich, da sie keine Files enthalten. Mit »R Sektornummer, Bufferanfang« liest man einen Sektor, entweder mit 128 oder 256 Byte Länge, in den Speicher ab »Bufferanfang«. Mit »W Sektornummer, Bufferanfang« läßt sich ein Sektor wieder zurück auf Diskette schreiben.

Hexadezimal-Dezimal-Umwandlung

Gibt man beispielsweise »? 100« ein, wird der entsprechende Hexadezimalwert ausgegeben. »? $100« wandelt eine Hexzahl in eine Dezimalzahl um.

Bildschirmdarstellung

Mit dem Befehl »K« steht normale oder invertierte Bildschirmdarstellung zur Wahl.

Maschinensprache-Unterprogramme aufrufen

Mit »G Adresse« kann ein Maschinensprache-Unterprogramm, das mit einer RTS-Anweisung enden muß, aufgerufen werden.



Der Programmcounter und die »+«-Variable

Zur Vereinfachung der Arbeit können Sie jederzeit die Platzhalter »*« und »+« verwenden. Dabei repräsentiert »*« den Programmcounter und wird in den Funktionen L, E, Z, A, D, U, F, und T mitgeführt. Ein mit der Funktion »L« gestartetes Listing setzt »L *« fort. »+« ist eine vom Benutzer frei definierbare Variable. Den Wert von »*« ändert man mit »* Wert«, und von »+« mit »+ Wert«. Die Befehle »* +« und »+ *« sind zulässig.

»Happy-Mon« arbeitet sowohl mit DOS-XL als auch mit DOS 2.0 oder DOS 2.5. Die abgedruckte Version läuft allerdings nur mit DOS-Versionen, die nicht ständig im Speicher verbleiben, wie DOS 2.0 oder DOS 2.5. Für Besitzer von DOS-XL oder anderen residenten DOS-Versionen befindet sich auf der Leserservice-Diskette eine zweite Version mit dem Namen »MASTER1.COM«.

Besondere Hinweise zu »Happy-Mon«

Geben Sie zunächst das Listing mit AMPEL ein. Ist »Happy-Mon« komplett abgetippt, speichern Sie das File bitte mehrfach. Legen Sie sich auch eine Kopie auf einer weiteren Diskette an, die Sie an einem sicheren Platz aufbewahren. Sollte dann Ihre Arbeitskopie versehentlich gelöscht werden, haben Sie immer noch eine Kopie. Sicher ist sicher!

Aufruf von »Happy-Mon«

Sollten Sie die abgedruckte Version verwenden (auf der Leserservice-Diskette finden Sie das File mit dem Namen »MASTER.COM«), müssen Sie entweder mit DOS 2.0 oder DOS 2.5 arbeiten. Um das Programm zu starten, gehen Sie zunächst mit »DOS« ins DOS-Menü. Dann laden Sie den Maschinensprache-Monitor mit der L-Funktion. Kehren Sie anschließend mit B zum Basic zurück, jetzt können Sie beispielsweise ein Basic-Programm eingeben oder von Diskette laden. »Happy-Mon« rufen Sie mit »? USR(32768)« auf. Sie befinden sich anschließend im Eingabeeditor des Maschinensprache-Monitors. Hier können Sie dann alle beschriebenen Funktionen ausprobieren. Mit X gelangen Sie zum DOS zurück und mit Q zum Basic. Sollte sich vor dem Aufruf von »Happy-Mon« ein Basic-Programm im Speicher befunden haben, steht einer Weiterbearbeitung nichts im Wege.

Die zweite »Happy-Mon«-Version ist nur auf der Leserservice-Diskette erhältlich. Sie finden Sie mit dem Namen »MASTER1.COM«. Diese Ausführung nutzt die Adressen ab $2200, also oberhalb von DOS-XL. Somit ist ein Zugriff auf die Adressen $8000 bis $9FFF, in dem sich »Happy-Mon« normalerweise befindet, möglich. Außerdem ist diese Version Voraussetzung, wenn Sie die Diskettenversion des MAC/65-Assemblers verwenden. Bevor Sie »Happy-Mon 2« einsetzen, ist noch das File »AUTRUN.A« in »AUTORUN.SYS« umzubenennen. Anschließend können Sie diese Diskette booten. Wenn Sie mit der Diskettenversion des MAC/65-Assemblers arbeiten, schalten Sie den Atari-Computer mit gedrückter OPTION-Taste ein, um das Basic zu desaktivieren. Von der DOS-Befehlszeile aus laden Sie dann anschließend das File »MASTER1.COM«. Ansonsten sind die Funktionen in beiden »Happy-Mon«-Versionen identisch.

Wichtige Anmerkung

»Happy-Mon« verwendet die Zeropage-Adressen 128 bis 160! Diese sollten nicht verändert werden. Sprünge von und zu Programmen, die diese Register benutzen, wie beispielsweise Basic und der MAC/65-Assembler, sind allerdings problemlos. »Happy-Mon« rettet diese Register bei der Initialisierung und stellt sie beim Verlassen (auch bei RESET) wieder her.

(Thomas Fischermann/wb)

Windows: Nicht nur ein Augenschmaus

Mit der Window-Technik läßt sich eine Benutzerführung realisieren, die bislang nur teuren Personal Computern vorbehalten war.

Hört man den Begriff »Window« oder Fenster, so verbindet man diese Begriffe mit einer hervorragenden Benutzerführung. In dieser Hinsicht hat besonders der Atari 520 ST von sich reden gemacht.

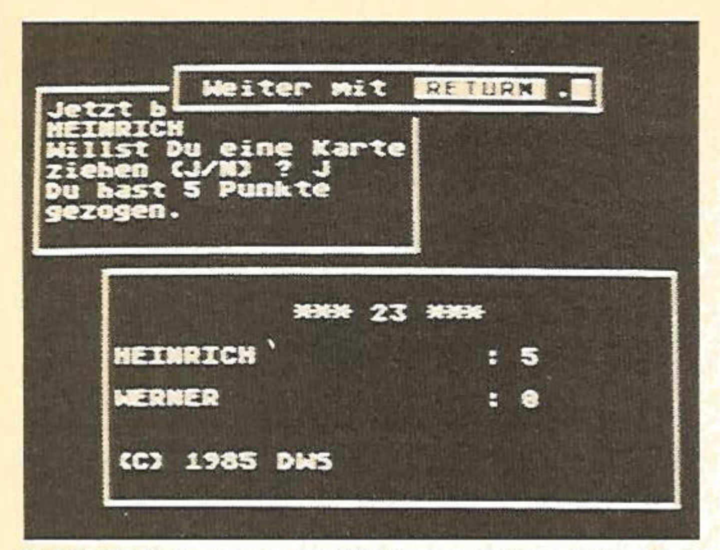
Das Programm »Windows« für den Atari 800XL/130XE ist sicher nicht mit GEM, der Benutzeroberfläche des Atari ST, zu vergleichen. Vielmehr soll es dem Atari-Fan zeigen, daß sich eine abgespeckte Version auch auf den »kleinen« Computern realisieren läßt. Mit diesem Programm kann man an einer beliebigen Stelle auf dem Bildschirm ein Fenster einblenden. Der Vorteil dieser »Window-Technik« liegt darin, verschiedenartige Texte oder Grafik und Text nebeneinander bearbeiten zu können. Löscht man das Fenster, erscheint auf dem Bildschirm wieder der ursprüngliche Text.

Der Trick dabei ist relativ einfach. Bevor man ein Fenster öffnet, wird der überblendete Bildschirmausschnitt in einen anderen Speicherbereich »gerettet«. Daraufhin ist dieser Bildschirmausschnitt für andere Arbeiten frei. Benötigt man das Fenster nicht mehr, wird der anfängliche Bildschirmausschnitt wieder in den Bildschirmspeicher geschrieben. Soviel zur Fenstertechnik. Wenden wir uns dem Programm selbst zu.

Tippen Sie zunächst Listing 1 mit dem Prüfsummer ab und speichern Sie dann das Programm. Nachdem Sie RUN eingegeben haben, wird ein File mit dem Namen WINDOWS.OBJ auf Diskettenlaufwerk Nummer eins abgelegt. Dieses File enthält dann die eigentlichen Window-Routinen. Um es einfach und schnell von Basic aus laden zu können, verwenden Sie Listing 2 (bitte mit dem Atari-Prüfsummer eingeben). Das Window-Initialisierungsprogramm liest dann die Window-Routinen in den Speicher. Da nun die Betriebssystemerweiterung im Speicher vorliegt, können Sie bedenkenlos mit NEW löschen.

Ein anderer Weg, das Programm »Windows« zu laden, ist von Basic aus »POKE 106,144:DOS« einzugeben. Vom DOS-

PROGRAMM-STECKBRIEF	
Programmname	Windows
Programmtyp	Utility
Programmiersprache	Basic und Maschinensprache
Programmlänge	4974 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	keine
Eingabehilfe	Prüfsummer und AMPEL
Bemerkung	Maschinenprogramm mit Basic-Lader
Leserservice	Diskette (LADE.BAS, WINDOWS.COM, WINDBEIS.BAS)



So könnte ein typischer, mit Windows versehener Bildschirm aussehen

Menü aus ruft man dann »WINDOWS.OBJ« auf und kehrt wieder ins Basic zurück. Nun noch »?USR(37120)« eintippen, und die Fenster sind abrufbereit. Die Betriebssystemerweiterung benötigt 4 KByte-RAM. Zieht man aber die Vorteile der Fenstertechnik in Betracht, dann ist dieser Speicher bestimmt kein verschwendeter Platz. Die Fenster werden in Grafikstufe 0 übrigens genauso angesprochen, wie andere Gerätetreiber auch.

Öffnen eines Windows für Ein- und Ausgabe

»POKE 1538,[X-Länge des Windows]: POKE 1539, [Y-Länge des Windows]: POSITION [X-Position], [Y-Position]: OPEN #[Kanalnummer 1-7],12,0,"W:"«.

Durch den OPEN-Befehl wird ein (zunächst leeres) Window auf den Bildschirm ausgegeben. Dabei enthalten die Speicherstellen 1538 und 1539 die X- und die Y-Ausdehnung (X-Ausdehnung = 5 bedeutet zum Beispiel, daß im Window 5 Spalten zur Verfügung stehen).

Beispiel: »POKE 1538,10:POKE 1539,8:POSITION 4,2:OPEN #1,12,0,"W:"«

Beschreiben eines Windows

»PRINT #[Kanalnummer wie bei OPEN]; "TEXT"«.
Die Ausgabe von Text erfolgt, wie man es von den Grafik-Modi 1 und 2 her kennt. Das ASCII-Zeichen 125 löscht den Inhalt des Windwos, und der windowinterne Cursor wandert dann wieder an die obere linke Ecke des Fensters. Wenn der Cursor dann den unteren Rand des Fensters erreicht hat, wird der Inhalt des Windows nach oben gescrollt. Beispiel: »PRINT #1;"5+7=";5+7«

Einlesen einer Textzeile

»INPUT #[Kanalnummer wie bei OPEN]; [Variablenname]«.
Der INPUT-Befehl wurde gegenüber dem üblichen INPUT etwas verbessert. Der Benutzer kann mit dem Cursor die INPUT-Zeile nicht mehr verlassen (zum Ausbessern von Fehleingaben stehen die Pfeiltasten und DELETE-BACKSPACE zur Verfügung). Außerdem kann man die maximale Länge der Eingabe festlegen, indem man den entsprechenden Wert in die Speicherzelle 844+[Kanalnummer*16] POKEt. Nach einem INPUT-Befehl steht in dieser Adresse wieder eine Null. Enthält diese Stelle keinen Wert größer Null, dann wird die Eingabelänge von der letzten Spalte des Windows begrenzt. Beispiel: »POKE 860,5:INPUT #1;A$« oder »INPUT #1;X«.

Positionieren des Cursors im Window

»X=[Spalte 0 bis X-Ausdehnung (Inhalt von 1538) des Windows minus 1]:Y=[Zeile 0 bis Y-Ausdehnung (Inhalt von 1539) des Windows minus 1]:POINT #[Kanalnummer wie bei OPEN],X,Y«.

Das nächste Zeichen wird dann an der entsprechenden Stelle ausgegeben (der POINT-Befehl funktioniert im Zusammenhang mit dem Window wie der POSITION-Befehl beim Bildschirm-Editor). Beispiel: »X=4: Y=1: POINT #1,X,Y«

Schließen eines Windows

»CLOSE #[Kanalnummer wie bei OPEN]«
Der ursprüngliche Inhalt des vom Window verdeckten Bildschirmteils wird wieder hergestellt. Beispiel: »CLOSE #1«.

Schließen eines Windows, ohne den ursprünglichen Bildschirminhalt wieder herzustellen

»POKE 832 +[Kanalnummer*16],255«. Nun kann der Kanal wieder benutzt werden, obwohl das Window auf dem Bildschirm bleibt. Beispiel: »POKE 848,255«.

Beim Öffnen und Schließen eines Windows (mit CLOSE) verlegt der Window-Handler den Cursor in die obere linke Ecke des Bildschirms.

Selbstverständlich erkennt der Window-Handler auch eventuelle Fehler (zum Beispiel Cursor außerhalb des Bildschirms). Man erhält dann die gewohnten Fehlermeldungen. Dadurch ist es fast unmöglich, die Window-Routinen durch falsche Werte zum Abstürzen zu bringen.

Wer noch einen Atari 400 oder 800 besitzt, muß in die Window-Initialisierungsroutine (Listing 2) die Zeile »32005 POKE 121,254:POKE 122,254« einfügen. Die INPUT-Routine des Window-Handlers benutzt nämlich diese Adresse (KEYDEF), die bei den XL- und XE-Modellen auf den Anfang der Tastaturbelegungstabelle zeigt, zur Tastaturabfrage. Durch Abändern von Listing 1 (Zeile 105 »OPEN #1,8,0,"C:"«) und Listing 2 (Zeile 32035 »OPEN #1,4,0, "C:":...«) läßt sich die Betriebssystemerweiterung an den Atari-Programmrekorder anpassen.

Listing 3 beinhaltet ein kleines Spiel, mit dem Sie die Window-Betriebssystemerweiterung ausprobieren können. Es handelt sich darum, durch Ziehen von Karten und Addition ihrer Einzelwerte möglichst nahe an die Zahl 23 heranzukommen, sie aber nicht zu überschreiten.

Windows - technisch betrachtet

Die zentrale Ein-/Ausgaberoutine des Betriebssystems (CIO) benutzt für die elementaren Ein-/Ausgabeoperationen wie Gerät öffnen, Gerät schließen, Byte lesen, Byte schreiben, Statusabfrage und sonstiges, die sogenannten Geräte-Handler. Die Anfangsadressen dieser Maschinensprach-Unterprogramme werden von der CIO über die Handler-Adressen-Tabelle (HATABS, $031A bis $033F) berechnet. Durch Ergänzen dieser Tabelle mit eigenen Handler-Adressen lassen sich beliebig eigene Ein-/Ausgabegeräte definieren.

(Dietrich Wagner/wb)

Initialisierungsprogramm (Window-Handler wird in HATABS eingetragen)	$9100-$9118
Window öffnen	$9119-$92A4
Window beschreiben	$92A5-$93B3
Sonstiges (Cursor positionieren)	$93B4-$93DF
Textzeile einlesen	$93E0-$961F
Zwischenspeicher für den vom Window verdeckten Bildschirminhalt	$9800-$9C99
Systemvariablen für den Window-Handler	$0600-$062F

Wichtige Adressen des Programms »Window«



```
0000:FF FF 00 91 1F 96 68 20<08>
0008:F6 95 B0 03 20 0B 96 60<67>
0010:18 91 37 92 DF 93 A4 92<62>
0018:1E 96 B3 93 4C 1C 96 20<3A>
0020:8B 92 A5 54 18 6D 03 06<B6>
0028:B0 06 30 04 C9 17 30 07<D1>
0030:A0 8D A9 FF 85 20 60 A5<9A>
0038:55 18 6D 02 06 B0 F1 30<B7>
0040:EF C9 27 30 03 4C 2A 91<F0>
0048:AD 02 06 F0 E3 AD 03 06<B4>
0050:F0 DE A6 54 A9 00 8D 00<60>
0058:06 8D 01 06 E0 00 F0 11<2F>
0060:AD 00 06 18 69 28 8D 00<FF>
0068:06 90 03 EE 01 06 CA D0<12>
0070:EF AD 00 06 18 65 55 8D<FF>
0078:00 06 90 03 EE 01 06 AE<49>
0080:03 06 E8 E8 AD 00 06 18<98>
0088:65 58 85 E0 AD 01 06 65<8A>
0090:59 85 E1 A5 58 A9 00 85<AF>
0098:E2 A9 98 85 E3 20 60 92<29>
00A0:A9 00 8D 10 06 8D 11 06<96>
00A8:A4 2E A9 00 99 4C 03 20<97>
00B0:B2 91 20 92 92 A0 01 60<90>
00B8:20 8B 92 AD 00 06 85 E0<CB>
00C0:AD 01 06 85 E1 A5 E0 18<DB>
00C8:65 58 85 E0 90 02 E6 E1<21>
00D0:A5 E1 18 65 59 85 E1 A0<E3>
00D8:00 A9 51 91 E0 20 22 92<8B>
00E0:C8 A9 45 91 E0 A9 00 8D<17>
00E8:04 06 EE 04 06 A0 00 A5<BE>
00F0:E0 18 69 28 85 E0 90 02<5D>
00F8:E6 E1 A9 7C 91 E0 20 2D<8A>
0100:92 C8 A9 7C 91 E0 AD 03<C4>
0108:06 CD 04 06 D0 DC A5 E0<40>
0110:18 69 28 85 E0 90 02 E6<F1>
0118:E1 A0 00 A9 5A 91 E0 20<3E>
0120:22 92 C8 A9 43 91 E0 60<AB>
0128:A9 52 C8 91 E0 CC 02 06<91>
0130:D0 F8 60 A9 00 C8 91 E0<BE>
0138:CC 02 06 D0 F8 60 20 8B<7A>
0140:92 AD 00 06 18 65 58 85<5E>
0148:E2 AD 01 06 65 59 85 E3<5B>
0150:A9 00 85 E0 A9 98 85 E1<E3>
0158:AE 03 06 E8 E8 20 60 92<43>
0160:20 92 92 A0 01 60 A0 FF<C8>
0168:C8 B1 E0 91 E2 CC 02 06<18>
0170:D0 F6 C8 B1 E0 91 E2 A5<95>
0178:E0 18 69 28 85 E0 90 02<E0>
0180:E6 E1 A5 E2 18 69 28 85<1C>
0188:E2 90 02 E6 E3 CA D0 D6<22>
0190:60 A0 00 A5 5D 91 5E 60<7B>
0198:A0 00 B1 58 85 5D 84 54<79>
01A0:84 55 A5 58 85 5E A5 59<84>
01A8:85 5F 60 C9 9B D0 03 4C<13>
01B0:29 93 C9 7D D0 0E 20 B2<C4>
01B8:91 A9 00 8D 10 06 8D 11<FA>
01C0:06 A0 01 60 8D 12 06 AE<6A>
01C8:01 06 AD 00 06 18 65 58<15>
01D0:A8 8A 65 59 AA 98 AC 11<63>
01D8:06 C8 18 69 28 90 01 E8<D2>
01E0:88 D0 F7 18 6D 10 06 90<42>
01E8:01 E8 18 69 01 90 01 E8<2F>
01F0:85 E0 86 E1 AD 12 06 29<EE>
01F8:7F C9 20 10 06 18 69 40<14>
0200:4C 04 93 C9 60 10 03 38<AB>
0208:E9 20 8D 13 06 A9 80 2C<6B>
0210:12 06 F0 09 AD 13 06 18<02>
0218:69 80 8D 13 06 AD 13 06<C2>
0220:A0 00 91 E0 EE 10 06 AD<E9>
0228:10 06 CD 02 06 D0 08 A9<97>
0230:00 8D 10 06 EE 11 06 AD<26>
0238:11 06 CD 03 06 30 03 20<37>
0240:3F 93 A0 01 60 AD 00 06<68>
0248:18 69 28 85 E2 AD 01 06<D1>
0250:69 00 85 E3 AD 00 06 18<F0>
0258:69 50 85 E0 AD 01 06 69<31>
0260:00 85 E1 A5 E2 18 65 58<8C>
0268:85 E2 A5 E3 65 59 85 E3<88>
0270:A5 E0 18 65 58 85 E0 A5<71>
0278:E1 65 59 85 E1 AE 03 06<57>
0280:CA F0 03 20 60 92 AE 03<EF>
0288:06 AD 00 06 18 65 58 85<E5>
0290:E2 AD 01 06 65 59 85 E3<F0>
0298:A5 E2 18 69 28 85 E2 90<18>
02A0:02 E6 E3 CA D0 F2 A9 00<B9>
02A8:A0 00 C8 91 E2 CC 02 06<57>
02B0:D0 F8 AE 03 06 CA 8E 11<07>
02B8:06 60 A6 2E BD 4C 03 30<7C>
02C0:1D CD 02 06 10 18 BC 4E<CB>
02C8:03 30 13 CC 03 06 10 0E<D7>
02D0:8D 10 06 8C 11 06 A9 00<8A>
02D8:9D 4C 03 A0 01 60 A9 00<75>
02E0:9D 4C 03 A0 8D 60 AD 29<D2>
02E8:06 F0 03 4C 9E 95 A9 01<9C>
02F0:8D 29 06 A4 2E AD 10 06<8A>
02F8:8D 20 06 8D 21 06 18 79<9D>
0300:4C 03 38 E9 01 30 10 CD<38>
0308:02 06 10 0B 8D 22 06 B9<F9>
0310:4C 03 F0 03 4C 18 94 AE<C3>
0318:02 06 CA 8E 22 06 A5 58<9D>
0320:18 6D 00 06 8D 23 06 A5<4F>
0328:59 6D 01 06 8D 24 06 AC<63>
0330:11 06 C8 AD 23 06 18 69<AE>
0338:28 8D 23 06 90 03 EE 24<0A>
0340:06 88 D0 EF AD 23 06 18<6A>
0348:6D 10 06 8D 23 06 AD 24<C0>
0350:06 69 00 8D 24 06 EE 23<C4>
0358:06 D0 03 EE 24 06 AD 11<38>
0360:06 8D 27 06 EE 02 06 AD<02>
0368:27 06 8D 11 06 20 23 95<51>
0370:A9 FF 8D FC 02 AD FC 02<BC>
0378:C9 FF F0 F9 AA 20 23 95<E1>
0380:8A C9 27 D0 03 4C 7B 95<D3>
0388:C9 3C D0 03 4C 89 95 A8<E2>
0390:B1 79 8D 26 06 C9 7D F0<CF>
0398:CE C9 7E D0 03 4C 3C 95<F9>
03A0:C9 1E D0 03 4C 59 95 C9<74>
03A8:9B F0 3E C9 1F F0 14 AE<4B>
03B0:BE 02 F0 0B C9 61 30 07<F3>
03B8:C9 7B 10 03 38 E9 20 18<B1>
03C0:6D B6 02 AE 20 06 8E 10<4A>
03C8:06 EC 22 06 30 08 F0 06<F8>
03D0:CE 20 06 18 90 0D C9 1F<29>
03D8:D0 03 4C 6E 95 20 A5 92<F8>
03E0:20 17 95 EE 20 06 4C 61<40>
03E8:94 A9 FF 8D FC 02 AD 21<FD>
03F0:06 8D 20 06 A9 00 A4 2E<25>
03F8:99 4C 03 AD 23 06 85 E2<A7>
0400:AD 24 06 85 E3 AD 22 06<09>
0408:38 ED 21 06 A8 B1 E2 D0<83>
0410:09 CE 22 06 88 10 F6 4C<69>
0418:E1 95 4C 9E 95 A0 80 8C<11>
0420:1F D0 8C 0A D4 C8 D0 F7<36>
0428:60 AD 23 06 85 E2 AD 24<75>
0430:06 85 E3 AD 20 06 38 ED<FA>
0438:21 06 A8 B1 E2 49 80 91<18>
0440:E2 60 AE 20 06 EC 21 06<F0>
0448:D0 03 4C 61 94 CA 8E 20<81>
0450:06 8E 10 06 A9 20 20 A5<F4>
0458:92 20 17 95 4C 61 94 AE<26>
0460:20 06 EC 21 06 D0 03 4C<4A>
0468:61 94 CA 8E 20 06 20 17<20>
0470:95 4C 61 94 AE 20 06 E8<0D>
0478:8E 20 06 20 17 95 4C 61<95>
0480:94 AD B6 02 49 80 8D B6<2C>
0488:02 20 17 95 4C 61 94 AD<79>
0490:BE 02 F0 05 A9 00 4C 95<39>
0498:95 A9 40 8D BE 02 20 17<B3>
04A0:95 4C 61 94 AD 20 06 CD<7A>
04A8:22 06 90 02 D0 39 38 ED<32>
04B0:21 06 A8 AD 23 06 85 E2<C6>
04B8:AD 24 06 85 E3 B1 E2 8D<67>
04C0:26 06 29 7F C9 40 B0 06<07>
04C8:18 69 20 4C CF 95 C9 60<BA>
04D0:B0 03 38 E9 40 A8 AD 26<22>
04D8:06 29 80 8C 26 06 0D 26<78>
04E0:06 EE 20 06 A0 01 60 A9<36>
04E8:00 8D 29 06 A9 9B CE 02<EE>
04F0:06 20 A5 92 20 17 95 A9<C1>
04F8:9B A0 01 60 A2 00 A0 21<89>
0500:BD 1A 03 D0 02 18 60 E8<E5>
0508:E8 E8 88 88 88 D0 F1 38<36>
0510:60 A9 57 9D 1A 03 A9 0A<A3>
0518:9D 1B 03 A9 91 9D 1C 03<DC>
0520:18 60 A0 01 60 60 00 00<96>
Laenge 1318 Bytes
```

Listing 1. Die »Windows«-Betriebssystemerweiterung. Bitte mit AMPEL eingeben.

```
32000 DATA▲104,104,141,85,3,104,141,84,3
,104,141,89,3,104,141,88,3,169,7,162,16,
32,86,228,96                              <NJ>
32010 RESTORE 32000:DIM BLOAD$(25)        <US>
32020 FOR F=1 TO 25:READ HI:BLOAD$(F,F)=
CHR$(HI):NEXT F                           <UA>
32030 POKE 106,144:GRAPHICS 0             <BU>
32035 OPEN #1,4,0,"D:WINDOS.OBJ":FOR F=1
 TO 6:GET #1,HI:NEXT F                    <CC>
32040 HI=USR(ADR(BLOAD$),37120,1400)      <QL>
32045 CLOSE #1                            <RM>
32050 HI=USR(37120)                       <TZ>
32060 END                                 <KS>
```

Listing 2. Basic-Lader für »Windows« (bitte mit dem Atari-Prüfsummer eingeben)

```
5 REM BEISP2.BAS                          <JA>
10 REM                                    <JO>
20 REM Zuerst "WINDINIT.BAS" laufen lass
en                                        <KL>
30 REM                                    <JQ>
100 REM *** 23 ***                        <IW>
150 ? "{ESC CTL <}":SETCOLOR 2,5,4:SETCO
LOR 1,0,14:SETCOLOR 4,5,0                 <TY>
200 DIM N$(40),I$(20),NA$(20),PU(2),AW$(
1),X(2),Y(2),CF(2)                        <CC>
210 N$="▲":N$(40)="▲":N$(2)=N$(1)         <SS>
220 X(1)=0:X(2)=18:PU(1)=0:PU(2)=0        <JT>
230 CF(1)=0:CF(2)=0                       <VC>
500 GOSUB 1200:REM >>> ANFANG <<<         <DV>
502 FOR N=1 TO 2                          <NU>
510 LX=20:LY=7:POSITION X(N),N*2-1:GOSUB
 2300                                     <MC>
520 GOSUB 1000                            <QQ>
530 IF CF(N)=1 THEN GOSUB 2100:GOTO 550   <EG>
540 GOSUB 2000                            <RF>
550 GOSUB 1600:REM >>> KARTE ZIEHEN <<<   <UR>
553 GOSUB 1800                            <UX>
555 GOSUB 1100:LX=22:LY=1:POSITION 8,0:G
OSUB 2300:? #3;"▲Weiter▲mit▲{CTL Y}RETUR
N{CTL Y}.";:INPUT #3;I$                   <GQ>
557 CLOSE #3                              <NE>
560 IF GEW>0 THEN GOSUB 1700:GOTO 502:RE
M >>> GEWONNEN <<<                        <JX>
570 NEXT N                                <IH>
580 GOTO 502                              <OM>
997 REM                                   <CD>
998 REM UP FRAGE                          <HS>
999 REM                                   <CJ>
1000 ? #3;"Jetzt▲bist▲Du▲dran,":? #3;N$(
N*20-19,N*20);                            <KS>
1010 ? #3;"Willst▲Du▲eine▲Karteziehen▲(J
/N)▲?▲";                                  <KG>
1020 RETURN                               <OZ>
1097 REM                                  <VB>
1098 REM UP PUNKTEANZEIGE                 <LE>
1099 REM                                  <VJ>
1100 LX=30:LY=10:POSITION 4,10:GOSUB 230
```

Listing 3. Ein einfaches Spiel mit Windows

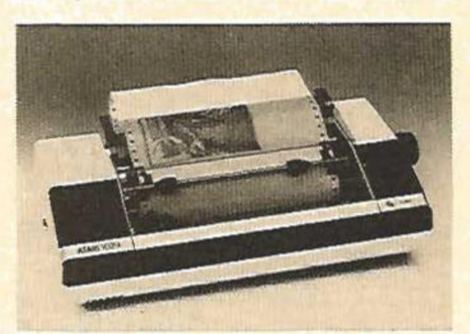
```
0                                              <EH>
1110 ? #3:? #3;"▴▴▴▴▴▴▴▴▴▴***▴23▴***"         <SI>
1120 ? #3:? #3;N$(1,20);":▴";PU(1):? #3:
? #3;N$(21,40);":▴";PU(2)                      <XM>
1130 ? #3:? #3:? #3;"(C)▴1985▴DWS"             <HC>
1140 RETURN                                    <PH>
1197 REM                                       <VD>
1198 REM UP ANFANG                             <TQ>
1199 REM                                       <VL>
1200 LX=32:LY=10:POSITION 3,0:GOSUB 2300       <ZU>
1210 ? #3:? #3;"▴▴▴▴▴▴▴▴▴▴▴***▴23▴***"        <QZ>
1230 ? #3:? #3;"Wie▴heisst▴der▴erste▴Spi
eler▴?":INPUT #3;I$:N$(1,20)=I$                <AJ>
1235 IF I$="" THEN N$(1,5)="ATARI":CF(1)
=1                                             <CF>
1240 ? #3:? #3;"Wie▴heisst▴der▴zweite▴Sp
ieler▴?":INPUT #3;I$:N$(21,40)=I$              <GA>
1245 IF I$="" THEN N$(21,28)="COMPUTER":
CF(2)=1                                        <ZK>
1250 CLOSE #3:RETURN                           <EQ>
1597 REM                                       <VL>
1598 REM UP KARTE ZIEHEN                       <LH>
1599 REM                                       <VT>
1600 IF  NOT KG THEN RETURN                    <PE>
1610 PP=INT(RND(0)*3)+3                        <ML>
1615 PU(N)=PU(N)+PP                            <TK>
1620 ? #3;"Du▴hast▴";PP;"▴Punkte":? #3;"
gezogen."                                      <SG>
1630 IF PU(N)>23 THEN ? #3;"Du▴hast▴verl
oren!":GEW=N+(N=1)-(N=2)                       <MH>
1635 IF PU(N)=23 THEN ? #3;"Du▴hast▴gewo
nnen!":GEW=N                                   <OH>
1640 RETURN                                    <PR>
1697 REM                                       <VN>
1698 REM UP ENDE                               <BE>
1699 REM                                       <VV>
1700 LX=34:LY=6:POSITION 2,2:GOSUB 2300        <UM>
1705 IF GEW=3 THEN ? #3;"Unentschieden!"
:? #3:GOTO 1730                                <MJ>
1710 ? #3;"Bravo,▴";N$(GEW*20-19,GEW*20)       <AK>
1720 ? #3;"Du▴hast▴gewonnen."                  <LM>
1730 ? #3:? #3;"Naechstes▴Spiel▴mit▴{CTL
 Y}RETURN{CTL Y}."                             <LO>
1740 INPUT #3;I$:PU(1)=0:PU(2)=0:GEW=0:C
LOSE #3                                        <ZR>
1750 RETURN                                    <PW>
1797 REM                                       <VP>
1798 REM PRUEFEN, OB SPIEL ZU ENDE             <JD>
1799 REM                                       <VX>
1800 IF PU(1)<21 OR PU(2)<21 OR GEW>0 TH
EN RETURN                                      <EE>
1810 IF PU(1)>PU(2) THEN GEW=1:RETURN          <ZH>
1820 IF PU(2)>PU(1) THEN GEW=2:RETURN          <AI>
1830 GEW=3:RETURN                              <AW>
1997 REM                                       <VT>
1998 REM UP SPIELERZUG                         <DS>
1999 REM                                       <WB>
2000 X=15:Y=3:POINT #3,X,Y:INPUT #3;AW$        <GJ>
2020 IF AW$="N" THEN KG=0:RETURN               <DN>
2030 IF AW$<>"J" THEN 2000                     <LY>
2040 KG=1:RETURN                               <AE>
2097 REM                                       <VC>
2098 REM UP COMPUTERZUG                        <EX>
2099 REM                                       <VK>
2100 GOSUB 2200                                <XY>
2110 IF KG=1 THEN ? #3;"J":RETURN              <IW>
2120 ? #3;"N":RETURN                           <BF>
2197 REM                                       <VE>
2198 REM UUP COMPUTERENTSCHEIDUNG              <DY>
2199 REM                                       <VM>
2200 IF PU(N)<19 THEN KG=1:RETURN              <XU>
2210 IF PU(N)>20 THEN KG=0:RETURN              <TA>
2220 IF PU(N+(N=1)-(N=2))<18 THEN KG=0:R
ETURN                                          <VZ>
2230 KG=1:RETURN                               <AF>
2297 REM                                       <VG>
2298 REM UUP WINDOW OEFFNEN                    <FA>
2299 REM                                       <VO>
2300 POKE 1538,LX:POKE 1539,LY:POKE 880,
255:OPEN #3,12,0,"W:":RETURN                   <YB>
```

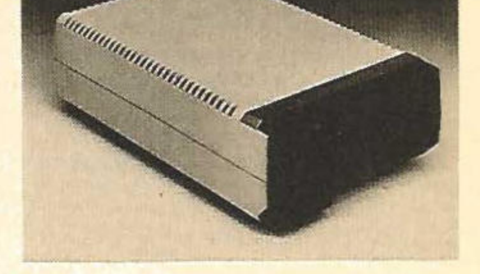
Listing 3. Ein einfaches Spiel mit Windows (Schluß)

Großes Preisausschreiben

Dieses Sonderheft haben wir für Sie gemacht, liebe Leser. Und damit Sie in Ihren Sonderheften nur solche Themen vorfinden, die Sie auch wirklich interessieren, brauchen wir Ihre Meinung. Schließlich soll ein mögliches nächstes Atari-Sonderheft noch besser, interessanter und informativer werden. Und damit wir wissen, was Sie wollen, schicken Sie bitte den ausgefüllten Fragebogen an uns zurück.

Natürlich gibt es auch etwas zu gewinnen. Unter allen Einsendungen verlosen wir viele interessante Preise.





- 2 Atari 1050-Diskettenlaufwerke, gestiftet von Atari
- 1 Atari 1029-Drucker, gestiftet von Atari
- 10 Gutscheine über je ein Buch freier Wahl aus dem Markt&Technik-Angebot
- 5 Textverarbeitungs-Programme Atari-Schreiber für alle Atari-Computer, gestiftet von Atari
- 20 T-Shirts im »Happy-Computer«-Look

Um an der Verlosung teilzunehmen, müssen Sie den folgenden Fragebogen ausfüllen, ihn in ein Kuvert stecken und an unsere Adresse schicken:

Redaktion Happy-Computer,
Markt&Technik Verlag AG,
Kennwort: Atari-Sonderheft,
Hans-Pinsel-Straße 2, 8013 Haar

Der Rechtsweg ist ausgeschlossen, Einsendeschluß ist der 30. April 1986.

(wb)



Fragebogen zum Atari-Sonderheft von Happy-Computer

Wie hat Ihnen dieses Sonderheft insgesamt gefallen?

☐ Sehr gut ☐ Gut ☐ Mittel ☐ Weniger gut ☐ Gar nicht

Wie fanden Sie die einzelnen Rubriken?
(zutreffendes bitte ankreuzen)

	Sehr interessant	interessant	uninteressant
Hardware-Basteleien	☐	☐	☐
Software-Tests	☐	☐	☐
Hardware-Tests	☐	☐	☐
Spiele-Test	☐	☐	☐
Anwendungs-Listings	☐	☐	☐
Spiele-Listings	☐	☐	☐
Tips und Tricks-Listings	☐	☐	☐
Turbo-Basic-Interpreter	☐	☐	☐
Turbo-Basic-Compiler	☐	☐	☐
Turbo-Basic-Teil (allgemein)	☐	☐	☐

Welche Beiträge haben Ihnen am besten gefallen?

1. ______
2. ______
3. ______
4. ______
5. ______
6. ______

Was haben Sie in diesem Sonderheft vermißt?

______

Wie oft würden Sie sich ein Sonderheft zum Thema Atari 800XL/130XE kaufen?

☐ Vierteljährlich ☐ Halbjährlich ☐ Einmal im Jahr

Welchen Computer besitzen Sie?

☐ 800 XL ☐ 130 XE ☐ 260 ST ☐ 520 ST+
☐ einen anderen, welchen? ______

Haben Sie vor, sich 1986 einen anderen Computer zu kaufen?

☐ ja ☐ nein

Wenn ja, welchen?

☐ 260 ST
☐ 520 ST+
☐ Amiga
☐ MS-DOS-Computer (z.B. IBM-PC, PC 10)
☐ ______

Möchten Sie sich einen Atari ST kaufen?

☐ ja ☐ nein

Falls nein, warum nicht? ______

Falls ja, warum? ______

Welche Peripherie besitzen Sie für Ihren Atari-Computer?

☐ Diskettenlaufwerk, Typ? ______
☐ Kassettenrecorder, Typ? ______
☐ Druckerinterface, Typ? ______
☐ Drucker, Typ? ______
☐ Monitor
○ S/W ○ Farbe Typ? ______
☐ Fernsehgerät
○ S/W ○ Farbe

Welche Peripherie wollen Sie sich 1986 noch kaufen?

☐ Diskettenlaufwerk, Typ? ______
☐ Kassettenrecorder, Typ? ______
☐ Druckerinterface, Typ? ______
☐ Drucker, Typ? ______
☐ Monitor
○ S/W ○ Farbe Typ? ______
☐ Fernsehgerät
○ S/W ○ Farbe

Welches Peripheriegerät würden Sie, falls Sie eines gewinnen, bevorzugen?

☐ Atari 1050-Diskettenlaufwerk
☐ Atari 1029-Drucker

Welche Computer-Zeitschriften lesen Sie regelmäßig?

______

Ich bin damit einverstanden, daß die hier gemachten Angaben elektronisch verarbeitet werden.

Anschrift:

______

Alter: ______ Jahre

Mein Atari-Computer



Zielgruppe: Einsteiger und Fortgeschrittene

Das Buch »Mein Atari-Computer« ist als Standardwerk für den Einsteiger anzusehen. Von den elf Kapiteln widmen sich fünf der Hardware. Es wird zunächst der Aufbau der älteren Modelle (Atari 400 und 800) besprochen; im Anhang findet man noch Erklärungen zum 800XL. Allerdings schenkt man den älteren Computern und Peripheriegeräten in diesem Kapitel allzuviel Aufmerksamkeit. Es werden teilweise Geräte beschrieben, die nicht mehr im Handel sind oder die es in Deutschland nie gegeben hat. Die neuen Computer kommen also zu kurz. Immerhin sind die Anschlüsse der Computer beschrieben, damit man weiß, welche Zusatzhardware sich an welchen Aus-/Eingang anschließen läßt.

Es lassen sich aber eine Reihe von Parallelen ziehen. Beispielsweise unterscheidet sich der Umgang mit dem alten Atari 810-Laufwerk nur in einigen wenigen Details vom neuen 1050-Laufwerk. Auch der Umgang mit dem alten Programmrecorder 410 entspricht dem mit dem 1010-Recorder. Man fühlt sich als Besitzer eines neuen Computer und Peripheriegerätes also nicht unbedingt vernachlässigt.

Interessiert sich jemand für Grafik, dann findet er in den Kapiteln »Einführung in die Grafikfunktionen des Atari-Computers« und »Weiterführende Beschreibung der Grafikfunktionen des Atari-Computers« wirklich ausführliche Informationen. In den beiden Kapiteln widmet sich der Autor auch der Player Missile-Programmierung, mit der bekanntlich bewegte Grafik erzeugt wird. Damit aber das Beschriebene noch verdeutlicht wird, hat man einige sehr gut dokumentierte Programmbeispiele miteingebaut. Schließlich läßt sich in der Praxis doch so manches besser nachvollziehen als mit bloßer Theorie.

Weiterhin beinhaltet das Buch auch noch einiges über Musik-Programmierung, Joystickabfrage und Paddles, das Diskettenformat und wie man sich eine eigene Dateiverwaltung aufbauen kann. Auch zu diesen Themen gibt es Programmbeispiele.

Das letzte Kapitel wendet sich schließlich jedem einzelnen Basic-Befehl und jeder einzelnen Basic-Funktion zu. Hier findet man auch Informationen zu den verschiedenen XIO-Funktionen, denen im Handbuch zum Computer kaum Aufmerksamkeit geschenkt wird.

Der Anhang informiert den interessierten Leser noch über eine Reihe von PEEKs und POKEs, die der etwas fortgeschrittenere Programmierer unbedingt kennen sollte. Auch die Speicheraufteilung der alten sowie der neuen Computer erläutert ein eigenes Kapitel.

Das Buch »Mein Atari-Computer« ist ein rundum gut aufgemachtes und informatives Buch. Bei der Arbeit stört allerdings sehr, daß kein Stichwortverzeichnis existiert. Sucht man bestimmte Informationen, muß man nicht selten das Buch von vorne bis hinten durchblättern. Es wäre wünschenswert, diesem Buch zumindest einige Seiten mit einem separaten Stichwortverzeichnis beizulegen. Vielleicht geht dieser Verbesserungswunsch ja schon bei der nächsten Neuauflage in Erfüllung. (wb)

Lon Poole, Martin McNiff, Steven Cook, »Mein Atari-Computer«, te-wi Verlag, ca. 470 Seiten, ISBN 3-921803-18-7, Preis 59 Mark

Das Atari-Programmierhandbuch



Zielgruppe: Einsteiger

Wenn Sie noch nie programmiert haben, aber einen Atari-Computer besitzen oder kaufen möchten, dann bietet sich »Das Atari-Programmierhandbuch« für den optimalen Einstieg an. So beschreiben die ersten Kapitel, was ein Programm überhaupt ist und wozu es dient. Langsam wird man dann an sein erstes Programm herangeführt.

Auch macht das Buch den Atari-Besitzer von vornherein mit der richtigen Programmiertechnik vertraut. Anhand von Flußdiagrammen zeigt die Autorin anschaulich den systematischen, überlegten Aufbau eines Programms. Solche Diagramme dienen schließlich auch dazu, Programme möglichst optimal zu gestalten. So spart man sich einige Wiederholungen und das Programm wird gleichzeitig überschaubarer.

Neben ausführlichen Erklärungen des Atari-Basic-Befehlssatzes und Erläuterungen zu den Fehlermeldungen, wird auch die Unterprogrammtechnik sowie die Stringverarbeitung angesprochen.

In »Das Atari-Programmierhandbuch« kommen Programmbeispiele nicht zu kurz. Insgesamt sind es 55. Man muß sie aber nicht unbedingt abtippen, da man sämtliche Programme auch auf Diskette beziehen kann. (wb)

Linda M. Schreiber, »Das Atari-Programmierhandbuch«, Markt & Technik, 403 Seiten, ISBN 3-89090-062-3, Preis 52 Mark

Das Atari-Buch, Band 2



Zielgruppe: Fortgeschrittene und Profis

Es ist naheliegend, daß man sich nach dem Kauf seines Atari-Computers zunächst mit der Basic-Programmierung beschäftigt. Schließlich liegt das Basic gleich nach dem Einschalten des Computers vor. Hat man seinen Computer unter Basic aber ausgereizt, steht Maschinensprache-Programmierung an. Schließlich ist nichts schneller als reine Maschinensprache. Hinzu kommt, daß sich gewisse Funktionen des Atari-Computers nur von Maschinensprache aus realisieren lassen.

Für »Das Atari-Buch, Band 2« werden zumindest Basic-Programmier-Kenntnisse vorausgesetzt. Sollten Sie sich mit Maschinensprache bereits ein wenig beschäftigt haben, ist dies auch vorteilhaft. Für die Einführung in die Maschinensprache-Programmierung sind am Ende des Buches ein Editor, ein Assembler und ein zugehöriger Disassembler abgedruckt. Somit kann man auch alle im Buch aufgelisteten Maschinenprogramme gleich ausprobieren.

In einem eigenständigen Kapitel werden die Interruptfähigkeiten der Atari-Computer erschöpfend behandelt. Mit dieser Technik lassen sich einige Programmvorgänge unabhängig vom restlichen Programm ausführen. Besonders wichtig ist die Interrupttechnik in Spielen.

Das letzte Kapitel widmet sich der Display-List, mit der man die Bildschirmausgabe

beeinflussen kann. Weiterhin wird in diesem Kapitel auch waagrechtes und senkrechtes Scrollen besprochen.

Wer also mehr aus seinen Programmen machen möchte, sollte auf »Das Atari-Buch, Band 2« zurückgreifen. Schließlich steckt im Atari mehr als nur ein eingebautes Basic. (wb)

Hans Lorenz Schneider, Rudolf Bichler, »Das Atari-Buch, Band 2«, Markt & Technik, 197 Seiten, ISBN 3-89090-072-0, Preis 32 Mark

Sprühende Ideen mit Atari-Graphik



Zielgruppe: Anfänger und Grafikinteressierte

Mit einer Auflösung von 320x192 Punkten und bis zu 256 gleichzeitig darstellbaren Farbschattierungen, ist der Atari-Computer kaum zu schlagen. Allerdings ist die Grafikprogrammierung nicht auf Anhieb verständlich. Vor allem wenn man seine Programme mit bewegter Grafik ausschmükken möchte, wird es kompliziert. Die Player Missile-Grafik, die für Bewegung auf dem Bildschirm sorgt, läßt sich nämlich nur mit entsprechenden Kenntnissen programmieren.

Wer animierte Grafik einsetzen, mehr über die verschiedenen Grafikmodi des Atari-Computers wissen, eine neue Display-List erzeugen und einen neuen Zeichensatz definieren möchte, findet zu diesen Themen jeweils ein eigenes Kapitel vor. Um Ihr neu angeeignetes Wissen auch gleich zu testen, wurden in das Buch Fragen eingebaut, die am Ende des Buches beantwortet werden.

Im Anhang finden sich noch einige Arbeitsblätter zum Entwurf und zur Gestaltung von Bildschirmen. Diese Seiten sollte man sich natürlich fotokopieren, um diese wirklich sinnvollen Hilfen immer wieder einzusetzen. (wb)

Tom Rowley, »Sprühende Ideen mit Atari Graphik«, te-wi Verlag, 211 Seiten, ISBN 3-921803-39-X, Preis 49 Mark

Atari 600XL/800XL Intern



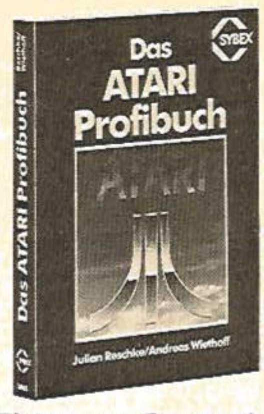
Zielgruppe: Fortgeschrittene und Profis

Wenn Sie sich bereits in Basic auskennen, sich anschließend der Maschinensprache zuwenden und gleichzeitig natürlich auch mehr über die Hardware des Atari-Computers wissen möchten, dann ist das Buch »Atari 600XL/800XL Intern« für Sie ein unerläßliches Hilfsmittel. Angefangen von der Speicherverteilung, bis hin zu ausführlichen Erklärungen der einzelnen Spezialbausteine des Atari-Computers, findet man alles Wissenswerte. Ausführlich wird jedes einzelne Register, jeder wichtige Baustein besprochen, sei es der Antic (zuständig für Grafik), der GTIA (Bildausgabe) oder der Pokey (zuständig für Tonausgabe, Tastaturabfrage etc.).

Im letzten Kapitel ist noch der Gesamtspeicherplan der Atari-Computer abgedruckt, der nochmals alle Adressen genau beschreibt. Wer also wirklich alles über seinen Atari wissen möchte, für den ist das Buch »Atari 600XL/800XL Intern« ein absolutes Muß. (wb)

Eichler, Grohmann, »Atari 600XL/800XL Intern«, Data Becker, 383 Seiten, ISBN 3-89011-053-3, Preis 49 Mark

Das Atari Profibuch



Zielgruppe: Fortgeschrittene und die es werden möchten

Abgesehen von der »etwas anderen Einleitung« mit viel Humor, verbergen sich im »Atari Profibuch« Informationen, auf die man zukünftig nicht mehr verzichten kann. In diesem Buch findet man alles über Speicheradressen, die speziellen Bausteine und die Besonderheiten des Atari.

Vor allem den Besonderheiten der Atari-Computer haben sich die Autoren in diesem Buch zugewandt. So ist ein Hauptaugenmerk auf die Funktion des Antic-Bausteins und die verschiedenen GTIA-Grafikmodi gerichtet. Auch Informationen darüber, wie man Bildschirmscrollen realisiert, wie man den Display List Interrupt verändern kann und wie Interrupts allgemein anzusprechen sind, fehlen nicht.

Wer seinem Atari-Computer schon immer mal besser klingende Töne entlocken wollte, findet zu diesem Thema ein eigenständiges Assembler-Programm vor. Und wen zuletzt noch die Anschlüsse der Atari-Computer interessieren, kann sich am Ende des Buches über eine Reihe von Bildern freuen, die jeden einzelnen Anschluß genauestens erklären. Sogar einen Schaltplan des 800XL kann das »Atari Profibuch« aufweisen. Mit diesem Buch werden einem also alle Tore, nicht nur zur Hardwarewelt, weit geöffnet. (wb)

Julian Reschke, Andreas Wiethoff, »Das Atari Profibuch«, Sybex Verlag, 289 Seiten, ISBN 3-88745-605-X, Preis 42 Mark

6502-Assembler-Kurs für Beginner



Zielgruppe: Fortgeschrittene

Schon das Äußere des Buches weicht von der Norm ab. Es ist kein starres, gebundenes Buch, sondern eher als Paperback zu bezeichnen. Es bezieht seine Angaben auf den Assembler T.EX.AS (= Terminal Extended Assembler), welcher ebenfalls aus dem Hause Dripke stammt.

Das Buch ist sowohl für die Theorie als auch für die Praxis geschrieben, so daß man es am besten am Computer bearbeitet. Die Assemblerbefehle werden in kleinen Programmen vorgestellt, so daß die Wirkung aller Befehle sofort ausprobiert werden kann.

Zwischen diesen mehr praktisch orientierten Kapiteln sind andere mehr theoretischer Art eingestreut, in welchen von Bits und Bytes, vom Speicheraufbau oder von den verschiedenen Adressierungsarten in der Assemblersprache gesprochen wird. Sie sind, wie das ganze Buch, in einer lockeren, leicht verständlichen Sprache geschrieben. Und hat man wirklich einmal etwas nicht verstanden, so kommt am Ende eines jeden Kapitels noch einmal eine Zusammenfassung, in welcher das Wichtigste der letzten Seiten wiederholt wird.

(Arnd Wängler)

Andreas Dripke, »6502-Assemblerkurs für Beginner«, Interface Age Verlag, ISBN 3-88986-000-1, Preis 29,80 Mark

Basic-Schalter

Möchte man das eingebaute Basic abschalten, muß man normalerweise den Atari-Computer erneut booten. Die Maschinencode-Routine »BSWITCH.COM« kann das genauso gut.

Als man sich bei Atari dazu entschieden hat, das eingebaute Basic durch Drücken der OPTION-Taste nur während des Bootens zu desaktivieren, hat man einige Anwender nicht gerade glücklich gemacht. Doch die Notwendigkeit ständig neu zu booten, besteht nun nicht mehr, denn mit der kurzen Maschinensprachroutine »BSWITCH.COM« kann man das Basic von DOS aus ein- beziehungsweise ausschalten.

Tippen Sie zunächst das kleine Basic-Programm ab, das die benötigte Datei mit dem Namen »BSWITCH.COM« auf der Diskette erzeugt. Sie können nun durch einfaches Laden des Programms »BSWITCH.COM« das eingebaute Basic ein- oder ausschalten. Geben Sie dazu unter DOS 2.5 zunächst L ein. Betätigen Sie anschließend die RETURN-Taste und laden Sie dann das File »BSWITCH.COM«. War das Basic zu diesem Zeitpunkt eingeschaltet, wird es ausgeblendet; war es ausgeschaltet, funktioniert das Ganze umgekehrt.

Wichtiger Hinweis: Basic funktioniert nur dann einwandfrei, wenn beim Booten nicht OPTION gedrückt wurde!

(Julian F. Reschke/wb)

```
1000 REM Basic Switch V 1.0                    <WW>
1010 REM von Julian F. Reschke                 <ZB>
1020 REM (c) Happy Computer                    <EV>
1030 REM                                       <TH>
1040 OPEN #2,8,0,"D:BSWITCH.COM"               <IN>
1050 PRUEF=0                                   <IS>
1060 FOR I=1 TO 23                             <GC>
1070 READ WERT:PRUEF=PRUEF+WERT                <JN>
1080 PUT #2,WERT:NEXT I                        <FZ>
1090 CLOSE #2:IF PRUEF<>2301 THEN ? "ERR
OR"                                            <PX>
1100 END                                       <PR>
1110 DATA 255,255,0,6,10,6,173,248,3,73,
1,141,248,3,76,116,228,224,2,225,2,0,6         <SD>
```

Basic-Listing »BSWITCH«

```
1000 ;------------------------
1010 ;Basic Switch V 1.0
1020 ;von Julian F. Reschke
1030 ;(c) Happy Computer 1985
1040 ;------------------------
1050 ;
1060 RUNAD = $02E0     ;Einsprungvektor
1070 BASICF = $03F8    ;Basic-Flag
1080 WARMSV = $E474    ;Warmstartvektor
1090 ;
1100      *= $0600
1110 ;
1120      LDA BASICF
1130      EOR #1        ;Status 'umdrehen'
1140      STA BASICF
1150      JMP WARMSV    ;Warmstart machen
1160 ;
1170      *= RUNAD
1180 ;
1190      .WORD $0600 ;Einsprungvektor
1200 ;
1210      .END
```

Quell-Listing »BSWITCH« (MAC/65-Assembler)

PROGRAMM-STECKBRIEF	
Programmname	Basic-Schalter
Programmtyp	Utility
Programmiersprache	Atari-Basic (Assembler)
Programmlänge	342 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	Prüfsummer
Bemerkung	Schaltet den Atari-Basic-Interpreter nach Belieben ein oder aus
Leserservice	Diskette (BSWITCH.BAS)

Blitzschnelle Zeichen-umwandlung

Die interne Zeichendarstellung unterscheidet sich vom normalen Zeichensatz. Eine Maschinenroutine liest den Bildschirm und ordnet die gelesenen Zeichen blitzschnell einem String zu.

Ein gemeinsames lästiges Übel der meisten gängigen Heimcomputer ist, daß, je nach Situation, eine andersartige Kodierung von Zeichen benutzt wird. Bei Atari-Computern muß man zwischen zwei verschiedenen Darstellungen unterscheiden.

Die ASCII-Darstellung wird für alle standardmäßigen Ein- und Ausgaberoutinen des Betriebssystems verwandt. Für den Basic-Programmierer bedeutet dies: Alle Basic-Kommandos wie PRINT, INPUT, LOCATE etc. arbeiten mit dieser Zeichendarstellung.

Daneben gibt es noch die interne Darstellung von Zeichen. Diese findet man beispielsweise bei der Reihenfolge der Zeichen in einem Zeichensatz oder bei der Kodierung der Zeichen im Bildschirmspeicher vor.

Gerade hier setzt das Programm »SCRSTR« an. Häufig gilt es, auf dem Bildschirm dargestellte Zeichen in eine normale Zeichenkette zu übertragen. Das Betriebssystem stellt zwar hierfür eine Funktion zur Verfügung, nämlich den LOCATE-Befehl, der aber auch Nachteile hat: Die einzelnen Zeichen müssen über eine Schleife von der Bildschirmposition in die Zeichenkette übertragen werden. Dies erfordert entsprechend viel Zeit. Solche Verzögerungen sollte man aber bei Programmen, die auf Tastatur-Eingaben reagieren, vermeiden.

Die Maschinenspracheroutine »SCRSTR« überträgt genau 960 Zeichen (40 Zeichen pro Zeile multipliziert mit 24 Zeilen ergibt 960 darstellbare Zeichen auf dem Bildschirm in Grafikstufe 0) eines normalen Textbildschirms in eine Zeichenkette. Diese muß vorher unbedingt auf die 960 Zeichen Länge dimensioniert werden. Teile des Bildschirms können anschließend problemlos als Teile dieser Zeichenkette behan-

delt werden. Hier ein Beispiel (SCREEN$ ist die benutzte Zeichenkette): Die zweite Bildschirmzeile erhält man mit »SCREEN$(41,80)«. Wird das 10. bis 20. Zeichen aus der 4. Zeile benötigt, so müssen die Zeichen in SCREEN$ als Position 130 bis einschließlich 140 gelesen werden.

Noch eine Anmerkung zum Basic-Programm. Nach dem Eintippen speichern Sie das Programm zunächst ab und geben RUN ein. Falls die Daten für die Maschinensprache-routine fehlerhaft sind, erfolgt eine entsprechende Meldung. Ansonsten wird als Demonstration das Programm geLISTet und in die Zeichenkette SCREEN$ übertragen, die daraufhin auf dem Bildschirm ausgegeben wird.

(Julian F. Reschke/wb)

PROGRAMM-STECKBRIEF	
Programmname	Screen to String
Programmtyp	Utility
Programmiersprache	Basic (Assembler)
Programmlänge	869 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	Prüfsummer
Bemerkung	Umwandlung von interner Darstellung in normalen Zeichensatz
Leserservice	Diskette (SCRSTR.BAS)

```
1000 REM -----------------------         <WX>
1010 REM Screen to String V 1.0          <ZU>
1020 REM von Julian F. Reschke           <ZE>
1030 REM (c) Happy Computer 1985         <FV>
1040 REM -----------------------         <XJ>
1050 REM                                 <TN>
1060 REM String fuer Bildinhalt          <KB>
1070 DIM SCREEN$(960):SCREEN$(960)="▲"   <VE>
1080 REM Bild fuellen                    <VW>
1090 POKE 82,0:LIST                      <WT>
1100 REM Routine definieren              <WY>
1110 DIM SCRSTR$(91):PRUEF=0             <LE>
1120 FOR I=1 TO 91                       <IX>
1130 READ WERT:SCRSTR$(I,I)=CHR$(WERT)   <LB>
1140 PRUEF=PRUEF+WERT:NEXT I             <PA>
1150 IF PRUEF<>12436 THEN ? "Fehler!":EN
D                                        <YV>
1160 DATA▲104,104,133,215,104,133,214,16
5,88,133,212,165,89,133,213,24,165,88,10
5,192,133,216,165,89,105                 <HC>
1170 DATA▲3,133,217,160,0,162,0,177,212,
16,2,162,255,41,127,201,64,16,5,24,105,3
2,208,9,201,96,16,5,56,233               <QP>
1180 DATA▲64,16,0,224,255,208,2,73,128,1
45,214,230,212,208,2,230,213,230,214,208
,2,230,215,165,212,197                   <LY>
1190 DATA▲216,208,200,165,213,197,217,20
8,194,96                                 <RM>
1200 REM                                 <TC>
1210 REM Routine aufrufen                <GF>
1220 WERT=USR(ADR(SCRSTR$),ADR(SCREEN$)) <QC>
1230 REM SCREEN$ ausgeben                <KT>
1240 PRINT SCREEN$                       <TH>
```

Basic-Programm zu »SCRSTR«

```
1000 ;-----------------------
1010 ;Screen to String V 1.0
1020 ;von Julian F. Reschke
1030 ;(c) Happy Computer 1985
1040 ;-----------------------
1050 ;
1060 SAVMSC = $58     ;Vektor auf Anfangs
adresse des Bildspeichers
1070 FR0 =   $D4
1080 DEST =  $D6      ;Zeiger auf String
1090 ENDE =  $D8      ;Zeiger auf Ende de
s Bildes
1100 ;
1110     *=  $0600    ;Programm ist versc
hiebbar
1120 ;
1130     PLA          ;Anzahl der Paramet
er
1140     PLA          ;Anfangsadresse des
 Strings
1150     STA DEST+1
1160     PLA
1170     STA DEST
1180 ;Adr. des Bildsp. holen
1190     LDA SAVMSC
1200     STA FR0
1210     LDA SAVMSC+1
1220     STA FR0+1
1230 ;Ende des Bildsp. berechnen
1240     CLC
1250     LDA SAVMSC
1260     ADC # <960
1270     STA ENDE
1280     LDA SAVMSC+1
1290     ADC # >960
1300     STA ENDE+1
1310 ;Hauptschleife
1320 CONVERT LDY #0
1330     LDX #0
1340     LDA (FR0),Y
1350     BPL NICHT_INVERS
1360     LDX #$FF
1370 NICHT_INVERS
1380     AND #$7F
1390     CMP #64
1400     BPL GR_64
1410     CLC
1420     ADC #32
1430     BNE FERTIG
1440 GR_64 CMP #96
1450     BPL FERTIG
1460     SEC
1470     SBC #64
1480     BPL FERTIG
1490 FERTIG
1500     CPX #$FF     ;Invers gewesen?
1510     BNE WAR_NICHT
1520     EOR #$80     ;Invertieren
1530 WAR_NICHT
1540     STA (DEST),Y
1550     INC FR0
1560     BNE KEIN_CARRY
1570     INC FR0+1
1580 KEIN_CARRY
1590     INC DEST
1600     BNE KEIN_CARRY2
1610     INC DEST+1
1620 KEIN_CARRY2
1630     LDA FR0      ;Ende?
1640     CMP ENDE
1650     BNE CONVERT
1660     LDA FR0+1
1670     CMP ENDE+1
1680     BNE CONVERT
1690 ;das war's
1700     RTS
1710 ;
1720     .END
```

Quell-Listing zu »SCRSTR« (MAC/65-Assembler)

Bildschirmausschnitt schnell gelöscht

Erfolgen häufig wechselnde Bildschirmausgaben, sollten bestimmte Ausschnitte des Monitors auch gleich wieder frei sein.

Manchmal ist es ganz nützlich, wenn man bestimmte Bildschirmbereiche schnell löschen kann. Zwar ist dies durchaus von Basic aus realisierbar, allerdings etwas langsam. Schneller geht es in Maschinensprache. Die kurze Routine »Bildloesch« läßt ein beliebig großes Feld auf dem Bildschirm im Bruchteil einer Sekunde verschwinden.

Die Maschinen-Unterroutine wird durch die Programmzeilen 1000 bis 1120 in den Speicherbereich ab 1536 gePOKEt. Hierbei handelt es sich um die Page 6, die besonders für kurze Maschinenroutinen geeignet ist. Der Aufruf des Programms erfolgt mit »X=USR(1536,SC-1+DS,ZEILEN,SPALTEN)«.

Dabei enthält SC die Bildschirmstart-Adresse. Von Basic aus läßt sich dieser Wert auch mit den Adressen 88 und 89 ermitteln. SD stellt die absolute Bildschirmposition dar. Sie wird mit Zeile x 40 + Spalte errechnet. ZEILEN gibt die Anzahl der Zeilen und SPALTEN die Anzahl der Spalten an, die das zu löschende Feld umfassen soll.

PROGRAMM-STECKBRIEF	
Programmname	Bildloesch
Programmtyp	Utility
Programmiersprache	Maschinensprache
Programmlänge	690 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	keine
Eingabehilfe	Prüfsummer
Bemerkung	löscht Teile des Bildschirms
Leserservice	Diskette (BLOESCH.BAS)

```
20 REM DER AUFRUF ERFOGT MIT
30 REM X=USR(1536,SC,ZEILEN,SPALTEN)
40 REM ** VON WOLFGANG CZERNY
90 GRAPHICS 0:LIST
100 GOSUB 1000
110 SC=PEEK(88)+256*PEEK(89)
120 ZEILE=10:SPALTE=5
125 DS=ZEILE*40+SPALTE
130 X=USR(1536,SC-1+DS,10,20)
140 GOTO 140
1000                         CH 1536
1010
1020                         EAD D:POKE A,D:
S=S+D
1030                         ATEN-FEHLER!":S
TOP
1090
1100                         ,104,133,176,10
4,104
1110                         45,176,136,208,
251,24,165,176,105,40,133
1120 DATA 176,165,177,105,0,133,177,202,
208,231,96
```

Hier ist bereits ein Teil des Bildschirms gelöscht

In der vorliegenden Form funktioniert die Routine »Bildloesch« nur in Grafikstufe 0. Weiterhin noch in den Grafikstufen, die 40 Byte pro Zeile in Anspruch nehmen. Ansonsten muß der vorletzte Wert in Zeile 1110 entsprechend angepaßt werden. Soll das Programm beispielsweise in Grafikstufe 1 verwendet werden, muß hier der Wert 20 anstatt 40 stehen.

(Wolfgang Czerny/wb)

```
10 REM LOESCHEN VON BILDTEILEN                <GS>
20 REM DER AUFRUF ERFOGT MIT                  <OM>
30 REM X=USR(1536,SC,ZEILEN,SPALTEN)          <KX>
40 REM ** VON WOLFGANG CZERNY                 <WO>
90 GRAPHICS 0:LIST                            <HS>
100 GOSUB 1000                                <QI>
110 SC=PEEK(88)+256*PEEK(89)                  <FS>
120 ZEILE=10:SPALTE=5                         <SO>
125 DS=ZEILE*40+SPALTE                        <EC>
130 X=USR(1536,SC-1+DS,10,20)                 <YJ>
140 GOTO 140                                  <NG>
1000 REM UNTERPROGRAMM NACH 1536              <WW>
1010 S=0:RESTORE 1100                         <OX>
1020 FOR A=1536 TO 1575:READ D:POKE A,D:
S=S+D:NEXT A                                  <LO>
1030 IF S<>5568 THEN ? "DATEN-FEHLER!":S
TOP                                           <WZ>
1090 RETURN                                   <PU>
1100 DATA 104,104,133,177,104,133,176,10
4,104,170,104,104,133,178                     <LZ>
1110 DATA 164,178,169,0,145,176,136,208,
251,24,165,176,105,40,133                     <KQ>
1120 DATA 176,165,177,105,0,133,177,202,
208,231,96                                    <AS>
```

Basic-Listing zu »Bildloesch«

```
0100 ;***** Loeschen einzelner
0110 ;***** Bildausschnitte Gr.0
0120 ;***** X=USR(1536,SCPOS,Y,X)
0130      *=  $0600
0140      PLA
0150      PLA          ;SCPOS NACH
0160      STA $B1      ;$00B0 UND
0170      PLA          ;$00B1
0180      STA $B0
0190      PLA          ;ZEILEN IN
0200      PLA          ;X=REG.
0210      TAX
0220      PLA          ;SPALTEN
0230      PLA          ;NACH $B2
0240      STA $B2
0250 L1   LDY $B2      ;$B2-YREG.
0260      LDA #0       ;LEERZEICHEN
0280 L2   STA ($B0),Y  ;SPEICHERN
0290      DEY          ;ZEILE
0300      BNE L2       ;ZU ENDE?
0310      CLC
0320      LDA $B0      ;ADDIERE
0330      ADC #$28     ;40 SPLATEN
0340      STA $B0      ;ZU SCPOS
0350      LDA $B1
0360      ADC #$00
0370      STA $B1
0380      DEX          ;ALLE ZEILEN
0390      BNE L1       ;FERTIG?
0400      RTS          ;RETURN
```

Quell-Listing zu »Bildloesch« (MAC/65)

Daten schnell zur Hand

Bilder, Daten und Variablen lassen sich mit »Turbo IO« blitzschnell von Diskette laden.

Es ist allgemein bekannt, daß im Betriebssystem des Atari-Computers sämtliche wichtigen Routinen für Ein- und Ausgaben bereits enthalten sind. Unter Basic ist die Auswahl allerdings beschränkt: Man kann die Befehle GET und PUT benutzen, die allerdings immer nur ein einzelnes Zeichen lesen beziehungsweise ausgeben. Eine Alternative stellen die Befehle PRINT und INPUT dar. Allerdings kann man mit diesen Befehlen nur maximal 125 Zeichen auf einen Massenspeicher schreiben und lesen.

Die Funktionen der CIO heißen nun »GET CHARACTERS« und »PUT CHARACTERS«. Sie sind zwar als die Befehle BGET und BPUT in Basic XL, Basic XE und im Turbo-Basic XL definiert, im Atari-Basic fehlen sie aber. Daher soll hier gezeigt werden, wie man sie über eine Maschinensprachunterroutine unter Atari-Basic ansprechen oder simulieren kann.

Für Ein- und Ausgaben stehen auf dem Atari acht »Kanäle« (0 bis 7) zur Verfügung. Zu jedem dieser Kanäle gibt es einen sogenannten IOCB (Input/Output-Control-Block). Für uns sind hier allerdings nur drei Register interessant.

ICCOM: Über dieses Register gibt man die Nummer des auszuführenden Befehls an. (Beispiel: »7« bedeutet »PUT CHARACTERS«, »11« bedeutet »GET CHARACTERS«.). Als nächstes interessiert uns die Zwei-Byte-Variable **ICBADR**, über die die Anfangsadresse des zu lesenden beziehungsweise zu schreibenden Speicherbereiches festgelegt wird. Zuletzt folgt noch **ICBLEN**. Hier gibt man die Anzahl der zu übertragenden Zeichen an.

Bevor man die Schnelladeroutine anspricht, muß ein Kanal bereits geöffnet sein. Jetzt folgt also nur noch der Sprung nach CIOV ($E456), und die gewünschte Funktion wird ausgeführt. Der Statuscode der Operation kann übrigens über das Y-Register abgefragt werden. Wir übertragen den Wert in die Variable FR0 ($D4) in Seite 0. Über die USR-Routinen können dann die Ergebnisse an das Basic-Programm zurückgegeben werden.

Falls nach RUN die Meldung »Fehler« ausgegeben wird, haben Sie einen Wert in den DATA-Zeilen falsch eingegeben. Korrigieren Sie in diesem Fall bitte das Programm.

Natürlich lassen sich auch Inhalte von Zeichenketten oder Zeichensätzen laden und speichern. Hier sind der Phantasie keine Grenzen gesetzt.

(Julian F. Reschke/wb)

PROGRAMM-STECKBRIEF	
Programmname	Turbo IO
Programmtyp	Utility
Programmiersprache	Atari-Basic (Assembler)
Programmlänge	1146 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	Prüfsummer
Bemerkung	Schnelles Laden und Speichern von Daten
Leserservice	Diskette (TURBOIO.BAS)

```
1000 REM ----------------------         <WX>
1010 REM Turbo-I/O V1.0                 <FB>
1020 REM von Julian F. Reschke          <ZE>
1030 REM (c) Happy Computer 1985        <FV>
1040 REM ----------------------         <XJ>
1050 DIM TURBO$(39)                     <JE>
1060 PRUEF=0                            <IV>
1070 FOR I=1 TO 39                      <LA>
1080 READ WERT:TURBO$(I,I)=CHR$(WERT)   <FX>
1090 PRUEF=PRUEF+WERT                   <VH>
1100 NEXT I                             <ER>
1110 IF PRUEF<>3595 THEN ? "Fehler!":END
                                        <MO>
1120 DATA 104,104,104,10,10,10,10,170,10
4,104,157,66,3,104,157,69,3,104,157,68  <XV>
1130 DATA 3,104,157,73,3,104,157,72,3,32
,86,228,132,212,169,0,133,213,96        <KD>
1140 REM                                <TM>
1150 REM Zur Demonstration:             <GY>
1160 GRAPHICS 24:SETCOLOR 2,0,0:COLOR 1 <AH>
1170 FOR I=0 TO 319 STEP 5              <MD>
1180 PLOT I,0:DRAWTO 319-I,191          <LC>
1190 NEXT I                             <FS>
1200 FOR I=0 TO 191 STEP 3              <HU>
1210 PLOT 0,I:DRAWTO 319,191-I          <PD>
1220 NEXT I                             <EZ>
1230 REM                                <TL>
1240 REM Bild abspeichern               <GI>
1250 REM Anfangsadresse: PEEK(88)+256*PE
EK(89)                                  <WP>
1260 REM Laenge: 192*40=7680            <LM>
1270 REM Befehl: Schreiben=11           <JC>
1280 OPEN #1,8,0,"D:TEST.PIC"           <QM>
1290 Q=USR(ADR(TURBO$),1,11,PEEK(88)+256
*PEEK(89),7680)                         <MG>
1300 CLOSE #1                           <MP>
1310 REM Bild loeschen                  <DL>
1320 GRAPHICS 24:SETCOLOR 2,0,0         <TP>
1330 REM Bild laden: Befehlscode 7      <NP>
1340 OPEN #1,4,0,"D:TEST.PIC"           <NY>
1350 Q=USR(ADR(TURBO$),1,7,PEEK(88)+256*
PEEK(89),7680):CLOSE #1                 <XU>
1360 GOTO 1360                          <RW>
```

Listing 1. »Turbo IO« als Basic-Lader

```
1000 ;--------------------
1010 ;Turbo-I/O V1.0
1020 ;von Julian F. Reschke
1030 ;(c) Happy Computer
1040 ;--------------------
1050 ;
1060 ;USR=(ADR(TURBO$),KANAL,KOMMANDO,AD
RESSE,LAENGE)
1070 ;
1080 FR0 =   $D4
1090 ICCOM = $0342
1100 ICBADR = $0344
1110 ICBLEN = $0348
1120 CIOV =  $E456
1130 ;
1140       *=  $0600
1150 ;
1160       PLA          ;Anzahl der Paramet
er
1170       PLA
1180       PLA          ;Kanalnummer
1190       ASL A        ;*16
1200       ASL A
1210       ASL A
1220       ASL A
1230       TAX
1240       PLA
1250       PLA          ;Kommandonummer
1260       STA ICCOM,X
1270       PLA          ;Adresse
1280       STA ICBADR+1,X
1290       PLA
1300       STA ICBADR,X
1310       PLA          ;Laenge
1320       STA ICBLEN+1,X
1330       PLA
1340       STA ICBLEN,X
1350       JSR CIOV     ;ab in's OS
1360       STY FR0      ;Fehlercode
1370       LDA #0
1380       STA FR0+1
1390       RTS
1400 ;
1410       .END
```

Listing 2. Quelltext zu »Turbo IO« (MAC/65-Assembler)

Wortumbruch perfekt

Ohne Rücksicht auf Verluste – sprich Trennregeln – werden unter Basic Worte einfach »zerrissen«. Mit der kurzen Routine »Wordwrap« passiert das nicht mehr.

Wer schon einmal versucht hat, ein Textadventure in Basic zu schreiben, kennt das Problem: Man kann zwar das Spiel auf dem Bildschirm verfolgen, aber einige Wörter werden am rechten Rand einfach »auseinandergerissen«. Diesem Manko setzt nun unsere kurze Routine »Wordwrap« ein Ende – und das, ohne beim Programmieren lange »herumzutüfteln«. Die kurze Maschinenroutine läuft auf allen XL- und XE-Modellen. Der auszugebende Text muß zunächst einmal in einem maximal 255 Zeichen langen String stehen. Ein String mit dem Namen TEXT$ soll nun auf dem Bildschirm dargestellt werden. Der Aufruf der »Wordwrap«-Routine erfolgt dann mit: »X=USR(Adresse,ADR(TEXT$), LEN(TEXT$))«. Der Wert »Adresse« bezieht sich dabei auf die Anfangsadresse der »Wordwrap«-Routine im Speicher (sie wird in einem String abgelegt).

Weiterhin kann man noch den rechten Rand mit »POKE 83,X« und den linken Rand mit »POKE 82,X« festlegen. Allerdings darf eine Zeile dann nicht kürzer als das längste verwendete Wort sein. (Bastian Robin/wb)

PROGRAMM-STECKBRIEF	
Programmname	Wortumbruch
Programmtyp	Utility
Programmiersprache	Maschinensprache
Programmlänge	1701 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	keine
Eingabehilfe	Prüfsummer
Bemerkung	Bei Textausgabe auf dem Bildschirm immer zusammenhängende Worte in jeder Zeile.
Leserservice	Diskette (WORDWRAP.BAS)

```
50 REM ******************************    <IC>
51 REM * Demo Programm fuer den     *    <RD>
52 REM * Wordprozessor. Eingabeteil *    <TT>
53 REM ******************************    <II>
100 DIM T$(255)                          <EE>
105 OPEN #1,12,0,"K:"                    <BZ>
106 POKE 62,10:POKE 83,30                <WB>
110 RESTORE :? "{ESC CTL <}":POSITION 5,
6:? "A)▲Text▲ohne▲RB▲Wordprozessor":POSI
TION 5,10:? "B)▲Text▲mit▲RB▲Wordprozesso
r"                                       <ZW>
120 GET #1,A:IF A=66 THEN 200            <EK>
130 IF A<>65 THEN 120                    <EL>
140 ? "{ESC CTL <}":? :FOR W=1 TO 9      <IH>
150 READ T$:? T$;:NEXT W                 <CW>
160 ? :? "{ESC CTL =}RETURN";:INPUT T$:G
OTO 110                                  <KM>
200 ? "{ESC CTL <}":? :FOR W=1 TO 9      <IA>
210 READ T$:GOSUB 1000:NEXT W            <XS>
220 GOTO 160                             <NZ>
990 REM *****************************    <MU>
991 REM * Das folgende 6502 Unter-  *    <WV>
992 REM * programm druckt den Text- *    <OE>
993 REM * string ohne Zerreissen    *    <SY>
994 REM * von Woertern Zeile fuer   *    <SK>
995 REM * Zeile auf den Bildschirm. *    <RZ>
996 REM * Laenge und Adresse des    *    <DV>
997 REM * Strings werden an das     *    <VQ>
998 REM * Unterprogramm uebergeben. *    <IX>
999 REM *****************************    <NV>
1000 A=ADR(T$):L=LEN(T$)                 <HE>
1040 X=USR(ADR("hh{CTL E}Lh{CTL E}KhhpG{
CTL E}M){CTL A}{CTL M}p{CTL B}▲{CTL ,}{C
TL D}N1KI▲p{CTL M}▲$qfN$NDMp+Pm"{ESC CTL
 >}hHDMp{CTL F}1KI▲Pt{CTL J}{CTL X}eUES{
CTL P}{CTL H}-GD▲$qfN$NDMp{CTL E}1K{CTL
X}{CTL P}J){CTL ,}{CTL M}p{CTL B}{CTL .}
"),A,L)                                  <BJ>
1050 RETURN                              <PI>
1990 REM ****************************    <RP>
1991 REM *    Data-Zeilen fuer      *    <GU>
1992 REM *    Adventure - Text      *    <GJ>
1993 REM ****************************    <SB>
2000 DATA▲Du▲stehst▲am▲Eingang▲einer▲gew
altigen▲Hoehle.▲Links▲neben▲dir▲fuehrt▲e
in▲Pfad▲nach▲Westen;▲er▲sieht            <NW>
2010 DATA▲,▲ziemlich▲verwildert▲aus▲und▲
wird▲von▲gewaltigen▲Grashalmen▲verdeckt.
▲Er▲scheint▲ins▲verwunschene▲Tal         <DW>
2020 DATA▲,▲zu▲fuehren.▲Richtung▲Osten▲e
rhebt▲sich▲ein▲gewaltiger▲Huegel▲der▲an▲
die▲Hoehle▲angrenzt.▲Ausserdem           <KA>
2030 DATA▲,▲dringen▲aus▲der▲Hoehle▲ploet
zlich▲merkwuerdige▲Laute;▲sie▲werden▲imm
er▲lauter▲und▲es▲hat▲den                 <EZ>
2040 DATA▲,▲Anschein▲als▲haetten▲sie▲Dic
h▲entdeckt....▲▲Was▲nun?                 <HY>
```

Listing zu »Wordwrap«

Hier sind die Texte am rechten Rand zerstückelt. Es fällt einem schwer, den Text zu lesen, da man am Ende einer Zeile den Rest des Wortes in der nächsten Zeile suchen muß.

Mit der Routine »Wordwrap« werden Worte am rechten Rand nicht abgeschnitten. Man kann den Text flüssig lesen. Nebenbei wirkt diese Darstellung auch viel professioneller.

Seitensprung ins Betriebssystem

Einige brauchbare Betriebssystem-Routinen lassen sich auch von Basic aus nutzen. Wir zeigen Ihnen wie.

Der Atari-Computer enthält insgesamt 24 KByte ROM, von denen das Basic genau ein Drittel, also 8 KByte einnimmt. Den anderen Teil beansprucht das Betriebssysten.

Dieses enthält alle wichtigen Unterprogramme, die der Computer schon nach dem Einschalten benötigt. So ist beispielsweise Basic zu aktivieren, sind Peripheriegeräte anzusteuern und ähnliches mehr. Das Basic selbst verwendet ständig Routinen des Betriebssystems – dies ist auch der Grund dafür, warum im Basic-Interpreter so viele Befehle zur Verfügung stehen. Es ist beachtlich, daß dafür nur 8 KByte ROM genügen.

Nichts, was mit Ein- und Ausgaben zu tun hat, wie beispielsweise die Befehle PRINT, GRAPHICS, INPUT oder OPEN erledigt das Basic selbst. Es spielen dabei immer Unterprogramme des Betriebssystems mit.

Da es die Atari-Computer bereits seit etwa sechs Jahren gibt und auch verschiedene Versionen (400/800/600XL/800XL/130XE) existieren, wurde das Betriebssystem mehrfach geändert. Wie soll man nun wissen, an welcher Stelle ein bestimmtes Unterprogramm zu finden ist?

Atari hat zum Zeitpunkt der Entwicklung des Betriebssystems vorausgeplant. So wurde dem Atari-Computer eine sogenannte »Sprungleiste« mit auf den Weg gegeben. Das bedeutet, daß man Unterprogramme nicht an ihrer eigentlichen Position im Speicher aufruft, sondern an einer festen Stelle in einer Tabelle von Einsprungvektoren, die sich garantiert nicht ändert. Ein Beispiel: Ob auf dem Atari 400 oder dem Atari 130 XE, mit einem Sprung nach Adresse $E477 (siehe Tabelle) wird ein Kaltstart ausgeführt. Bei $E477 selbst steht nur ein Sprungbefehl zur eigentlichen Kaltstart-Routine. Wo sich diese befindet, muß der Programmierer nicht unbedingt wissen.

Dieser Aufbau hat noch einen großen Vorteil: Wer sich nämlich ein anderes Betriebssystem-ROM in seinen Computer einbauen läßt oder sich ein modifiziertes Betriebssystem in den RAM-Bereich von $C000 bis $FFFF lädt, kann davon ausgehen, daß alle Programme, die sich an die Sprungtabelle halten, auch auf Ihrer Betriebssystemversion laufen. So kann man mit einem veränderten Betriebssystem beispielsweise einen RAM-Disk-Treiber für den 130 XE einbauen, der dann nicht nur unter DOS 2.5, sondern mit allen Programmen mit Diskettenzugriff funktioniert. Ein solcher Treiber für den zusätzlichen Speicher ist in verschiedenen Spielen bereits integriert.

Betrachten wir die Tabelle der Einsprungadressen, die auf den ersten Blick ziemlich kurz aussieht. Dies liegt ausschließlich daran, daß einige der Routinen fast universell einsetzbar sind. Die Sprungleiste beginnt bei der Adresse 58448 ($E450), und jede Einsprungstelle belegt natürlich 3 Byte (nämlich 1 Byte für einen Maschinensprache-»JMP« (Sprungbefehl) und 2 Byte für die Adresse). Jede Einsprungadresse hat eine feste Bezeichnung, die von Atari in den entsprechenden technischen Dokumentationen vorgegeben wurde. Ideal wäre es, wenn sich jeder an diese Benennungen halten würde. Dann wäre eine einheitliche Namensgebung gewährleistet und andere Programmierer könnten sich schneller in fremde Programme hineindenken. Jeder Name eines Einsprungs hat als letzten Buchstaben ein »v«, als Erinnerung daran, daß es sich ja nur um einen Vektor handelt, und nicht etwa um die tatsächliche Adresse der Routine im Betriebssystem-ROM.

Fünf Sprünge in die Tiefe

Der zweite Teil der Liste enthält fünf Einsprünge, die man nur auf dem 600 XL, 800 XL und 130 XE nutzen sollte. Programme, die auf dem Atari 400 und 800 laufen sollen, dürfen diese Einsprünge auf keinen Fall verwenden! Man sollte sich also folgendes merken: Programme, die für alle Geräteversionen bestimmt sind, dürfen nur Einsprungadressen zwischen $E450 und $E47F benutzen. Alle anderen Aufrufe von Routinen im Bereich ab $C000 aufwärts sind illegal.

Nach diesen Vorüberlegungen kommen wir zu den einzelnen Routinen. Hier muß man einerseits zwischen den Routinen unterscheiden, die für den Basic-Programmierer interessant sind und andererseits zwischen denen, die nur auf Maschinenspracheebene von Bedeutung sind. Beginnen wir mit einer kurzen Übersicht über die einzelnen Funktionen:

58448 $E450 – DSKIV

Über diese Routine kann man den seriellen Bus für die Benutzung der Diskettenstation initialisieren.

58451 $E453 – DSKINV

Ein- und Ausgaberoutine für die Diskettenstation, über die man einzelne Diskettenoperationen ausführen kann (beispielsweise Sektoren lesen oder schreiben).

58454 $E456 – CIOV

Dies ist der Einsprung für das wichtigste Unterprogramm des Betriebssystems, der CIO. Über die CIO (Central Input/Output Utility) können alle standardmäßigen Ein- und Ausgabeoperationen durchgeführt werden (Datei öffnen, Bytes schreiben, Bytes lesen etc.). Bis auf zwei Befehle nämlich, BPUT und BGET (siehe auch den Artikel über eine Schnell-Laderoutine in diesem Sonderheft), sind alle CIO-Funktionen auch über Basic-Befehle ansprechbar.

58457 $E459 – SIOV

Einsprung in die Ein- und Ausgaberoutine für den seriellen Bus. Mit dieser Routine kann man alle Ein- und Ausgaben über den seriellen Bus durchführen, unter anderem natürlich auch die unter DSKINV aufrufbaren Funktionen.

58460 $E45F – SETVBV

Diese Routine kann Vektoren von wichtigen Betriebssystemroutinen ändern. Dafür ist eine eigene Routine nötig. Wird nämlich ein Vektor in dem Moment aufgerufen, in dem man ihn verändert, kann dies einen Systemabsturz zur Folge haben. Um einen Vektor zu ändern, muß man zunächst das X- und das Y-Register (High- und Low-Byte) mit dem gewünschten Wert laden. Mit dem Akku legt man dann den zu verändernden Vektor beziehungsweise Zähler fest:

Wert	Name	Bedeutung
0	VIMIRQ	IRQ-Vektor
1–5	CDTMVI-5	Systemzähler 1–5
6	VVBLKI	Vektor auf Immediate-VBI
7	VVBLKD	Vektor auf Deferred-VBI

58463 $E45F – SYSVBV

Routine zur Beendigung des Immediate-VBI (Vertical Blank Interrupt).

58466 $E462 - XITVBV
Routine zur Beendigung des Deferred-VBI.
58469 $E465 - SIOINV
Routine zur Initialisierung des Pokey-Chips, der beim Einschalten nicht initialisiert wird.
58472 $E468 - SENDEV
Pokey-Chip auf Ausgabe stellen.
58475 $E468 - INITIV
NMI-Interrupts initialisieren.
58478 $E46E - CIOINV
Routine zur Initialisierung der CIO (siehe oben).
58481 $E471 - BLKBDV
Vektor für den Basic-Befehl BYE, der beim Atari 400/800 in den Memo-Pad-Modus und bei den neueren Geräten in den Selbsttest führt.
58484 $E474 - WARMSV
Vektor für den Warmstart; entspricht dem Drücken der RESET-Taste.
58487 $E477 - COLDSV
Vektor für den Kaltstart; entspricht dem Aus- und Einschalten des Computers (allerdings wird der zusätzliche RAM-Bereich des XE nicht gelöscht).
58490 $E47A - RBLOKV
Einen Block vom Kassetten-Recorder einlesen.
58493 $E47D - CSOPIV
Einen Kanal für das Lesen von Daten vom Kassetten-Recorder öffnen.
58496 $E480 - PUPDIV
Sprung zum Einschaltbild, das beim XL und XE dem Selbsttest entspricht.
58499 $E483 - SLFTSV
Sprung in den Selbsttest.
58502 $E486 - PHENTV
Eintrag in der Gerätetreibertabelle vornehmen.
58505 $E489 - PHULNV und
58508 $E48C - PHINIV
Spezielle Routinen für die Benutzung des parallelen Busses.

Die Erfahrung zeigt, daß von den aufgeführten Routinen nur acht häufiger benutzt werden. Dies sind natürlich zunächst einmal die Einsprünge für Warmstart (von Basic: USR(58484)) und Kaltstart (USR(58487)). Dazu kommen SETVBV, SYSVBV und XITVBV, die bei Interrupt-Routinen des öfteren notwendig sind. Es bleiben die drei Ein- und Ausgaberoutinen DSKINV, CIOV und SIOV.

Da alle CIO-Funktionen (bis auf die oben genannte Ausnahme) direkt über Basic-Befehle ansprechbar sind, bleiben also für den Basic-Programmierer noch die beiden Routinen DSKINV und SIOV.

Sie steuern die Ein- und Ausgaben über den seriellen Bus, wobei DSKINV sozusagen eine auf die Diskettenstation »spezialisierte« Version von SIOV ist.

In der obigen Liste war sehr gut zu erkennen, daß man beim Aufruf von Betriebssystemroutinen die Prozessorregister oft auf bestimmte Werte setzen muß. Dies ist jedoch mit der USR-Funktion nicht möglich. Außerdem kann man mit USR nur speziell für Basic geschriebene Routinen aufrufen, da die mit USR übergebenen Werte automatisch auf dem Prozessor-Stack abgelegt werden. Daher benötigt man eine spezielle USR-Routine, die zunächst die Prozessorregister lädt und dann einen »echten« Sprung ausführt. Ein solches Programm finden Sie im abgedruckten Basic-Programm.

Bitte speichern Sie das Programm ab, bevor Sie es mit RUN ausprobieren.

Zunächst wird das 35 Byte lange Maschinenprogramm in die Zeichenkette CALL$ gelesen. Daraufhin wird zu Demonstrationszwecken, mit Hilfe der Betriebssystemroutine DSKINV, der Bildschirm in Grafikstufe 24 mit dem Inhalt der ersten 60 Sektoren auf der Diskette gefüllt. Hier eine Erklärung der Register, die man setzen muß:

Trickreiche Adressen

769 $301 - DUNIT
Nummer des anzusprechenden Diskettenlaufwerks (1 bis 8).
770 $302 - DCOMND
Kommando der auszuführenden Funktion. Das 1050-Laufwerk von Atari versteht folgende Kommandos:
33 (»!«): Formatieren mit 720 Sektoren
34 (»"«): Formatieren mit 1040 Sektoren
80 (»P«): Sektor schreiben ohne Überprüfung (läuft nur auf dem Atari 800, XL und 130 XE)
82 (»R«): Sektor lesen
83 (»S«): Status überprüfen
87 (»W«): Sektor schreiben mit Überprüfung
772,773 $304,$305 - DBUFLO,HI
Zeiger auf die Stelle im Speicher, ab der die Informationen eines Sektors von Diskette gelesen oder geschrieben werden.
778,779 $30A,$30B - DAUX1/2
Nummer des anzusprechenden Diskettensektors.

Betrachten wir nun den Rest des Beispielprogramms: In den Zeilen 1160 bis 1180 werden den Registern zunächst Konstanten zugeordnet. Dies ist nicht unbedingt notwendig, trägt aber zur Verständlichkeit des Programms bei. Es folgt in Zeile 1190 das Einschalten der hochauflösenden Grafik. Außerdem wird die Anfangsadresse des Bildspeichers berechnet und in der Variablen SC abgelegt. Bereits in Zeile 1200 werden die ersten Register für die DSKINV-Routine gesetzt: DUNIT enthält den Wert 1, weil das erste Laufwerk benutzt werden soll. DCOMND wird auf Lesen von Sektoren (»R« wie Read) gesetzt. Werte, die über den Bereich von 0 bis 255 hinausgehen, erscheinen beim Atari immer als Doppelbytes, bei denen das zweite Byte das höherwertige ist. Konkret heißt das, daß man den Wert

Tabelle der wichtigsten Einsprungvektoren ▼

Adresse			
Hex.	Dez.	Name	Beschreibung
$E450	58448	DSKIV	Initialisierung des Laufwerks
$E453	58451	DSKINV	Ein- und Ausgabe für Diskettenstation
$E456	58454	CIOV	Central Input/Output-Routine (CIO)
$E459	58457	SIOV	Serial Input/Output-Routine (SIO)
$E45C	58460	SETVBV	Interrupt-Vektoren setzen
$E45F	58463	SYSVBV	Beende Immediate-VBI
$E462	58466	XITVBV	Beende Deferred-VBI
$E465	58469	SIOINV	Initialisiere POKEY-Chip
$E468	58472	SENDEV	Schalte POKEY auf Ausgabe
$E46B	58475	INTINV	Interrupts initialisieren
$E46E	58478	CIOINV	CIO initialisieren
$E471	58481	BLKBDV	Vektor für »BYE«
$E474	58484	WARMSV	Warmstart-Vektor
$E477	58487	COLDSV	Kaltstart-Vektor
$E47A	58490	RBLOKV	Record von Kassette lesen

Die folgenden Einsprünge gibt es nur auf dem 600 XL, 800 XL und 130 XE!

Hex.	Dez.	Name	Beschreibung
$E480	58496	PUPDIV	Vektor zum Einschaltbild
$E483	58499	SLFTSV	Vektor zum Selbsttest
$E486	58502	PHENTV	Eintrag in HATABS vornehmen
$E489	58505	PHULNV	Eintrag löschen
$E48C	58508	PHINIV	Parallelen Bus initialisieren



einer solchen Adresse folgendermaßen berechnen kann:
»PEEK(Adresse)+256*PEEK (Adresse+1)«

Analog muß man natürlich vorgehen, wenn man ein solches 16-Bit-Register verändern will. Da nur die Sektoren bis Nummer 60 angesprochen werden sollen, muß das High-Byte der Sektornummer immer den Wert 0 enthalten. Es muß also nur ein einziges Mal, vor dem Eintritt in die Hauptschleife, richtig gesetzt werden. Anschließend noch mit der gleichen Methode DBUFLO/HI auf den Anfang des Bildspeichers bringen. Die Schleife von 1220 bis 1270 benutzt das Indexregister »I« als Schleifenzähler, so daß man I als Low-Byte der Sektornummer betrachten und in DAUX1 übertragen kann.

Schließlich folgt noch der Aufruf der Betriebssystemroutine. Da DSKINV keine speziellen Werte in den Prozessorregistern erwartet, wird hier einfach jeweils die 0 als Wert angegeben. Zu guter Letzt muß noch das Doppelbyte DBUFLO/HI um genau 128 erhöht werden, da schließlich jeder Sektor 128 Byte lang ist.

Sie haben sicherlich bemerkt, daß dabei nicht unbedingt die größte Ladegeschwindigkeit erreicht wird. Das liegt daran, daß die Basic-Befehle zwischen den einzelnen Aufrufen von DSKINV immer so lange warten müssen, bis die Diskette wieder eine volle Umdrehung zurückgelegt hat. Erst dann kann der nächste Sektor gelesen werden.

```
1000 REM --------------------         <RW>
1010 REM Betriebssystem-Demo          <QT>
1020 REM von Julian F. Reschke        <ZE>
1030 REM (c) Happy Computer           <EY>
1040 REM --------------------         <SI>
1050 DIM CALL$(35):CALL$(35)="▲"      <FS>
1060 PRUEF=0                          <IV>
1070 FOR I=1 TO 35                    <IG>
1080 READ WERT:PRUEF=PRUEF+WERT       <JQ>
1090 CALL$(I,I)=CHR$(WERT)            <JA>
1100 NEXT I                           <ER>
1110 IF PRUEF<>4907 THEN PRINT CHR$(125)
;"Datenfehler!":END                   <DD>
1120 REM                              <TG>
1130 DATA▲104,104,104,133,212,104,104,13
3,213,104,104,133,214,104,133,215,104,56
,233                                  <BD>
1140 DATA▲1,170,165,215,233,0,72,138,72,
165,212,166,213,164,214,96            <OJ>
1150 REM                              <TP>
1160 DUNIT=769:DCOMND=770             <QR>
1170 DBUFLO=772:DBUFHI=773            <YN>
1180 DAUX1=778:DAUX2=779:DSKINV=58451 <XB>
1190 GRAPHICS 24:SETCOLOR 2,0,0:SC=PEEK(
88)+256*PEEK(89)                      <EX>
1200 POKE DUNIT,1:POKE DCOMND,ASC("R"):P
OKE DAUX2,0                           <NQ>
1210 POKE DBUFHI,INT(SC/256):POKE DBUFLO
,SC-256*PEEK(DBUFHI)                  <IK>
1220 FOR I=1 TO 60                    <GI>
1230 POKE DAUX1,I                     <LX>
1240 WERT=USR(ADR(CALL$),0,0,0,DSKINV) <LV>
1250 LO=128+PEEK(DBUFLO):IF LO<256 THEN
POKE DBUFLO,LO:GOTO 1270              <SL>
1260 LO=LO-256:POKE DBUFHI,1+PEEK(DBUFHI
):POKE DBUFLO,LO                      <JQ>
1270 NEXT I                           <FO>
1280 GOTO 1280                        <SO>
```

Mit diesem Programm können von Basic aus Betriebssystem-Vektoren angesprungen werden. Anschließend kehrt man wieder ins Basic zurück.

DPEEKs und DPOKEs mit Turbo-Basic XL

Abhilfe schafft Turbo-Basic XL, das einerseits das Atari-Basic an Geschwindigkeit übertrifft, außerdem für die Veränderung von 16-Bit-Registern die schnelleren (und komfortableren) Befehle DPEEK und DPOKE zur Verfügung stellt.

Ungefähr das gleiche kann man mit der Routine SIOV ($E459) erledigen. Allerdings gibt es auch einige Unterschiede. Da man nicht auf die Diskettenstation festgelegt ist, sondern auch Drucker oder Kassetten-Recorder ansprechen kann, müssen einige Register zusätzlich initialisiert werden (diese Arbeit nimmt einem normalerweise der Einsprungvektor DSKINV ab);

768 $300 - DDEVIC

Mit diesem Register wird der anzusprechende Gerätetyp ausgewählt:
49 = Diskettenstation
64 = Drucker
96 = Kassetten-Recorder

771 $303 - DSTATS

Einerseits muß vor dem Aufruf von SIOV festgelegt werden, ob es sich um eine Ausgabe (128) oder eine Eingabe (64) handelt. Andererseits enthält DSTATS im Anschluß daran den Statuscode des seriellen Busses (entspricht normalerweise dem Basic-Fehlercode).

774 $306 - DTIMLO

Auch die Zeit in Sekunden, die die Ausführung eines Kommandos dauern darf, muß festgelegt werden. Für Diskettenoperation kann man eine 7 wählen (Ausnahme ist das Formatieren, für das man den Wert 160 ansetzen sollte). Für Operationen mit einem Kassetten-Recorder oder einem Drucker empfehlen sich Werte um 30.

776,777 $308,$309 - DBYTLO/HI

Anzahl der zu übertragenden Bytes. Bei der Diskettenstation normalerweise 128.

Bleibt anzumerken, daß der Befehl P (Schreiben ohne Überprüfung) auf dem Atari 400/800 voraussetzt, »SIOV« zu benutzen, da bei den alten Betriebssystemversionen dieser Befehl von DSKINV nicht unterstützt wurde.

Die Routine CALL$ können Sie übrigens auch verwenden, um eigene Maschinenspracheroutinen von Basic aus aufzurufen. Dies ist immer dann notwendig, wenn die betreffenden Routinen nicht als USR-Routinen für Basic konzipiert wurden.

Es zeigt sich also, daß man mit Einsprungadressen ins Betriebssystem der Atari-Computer viele wirklich nützliche Funktionen ausführen kann. Für den Basic-Programmierer öffnen sich somit viele sonst verschlossene Türen. Denn leider ist Basic in so manch einer Hinsicht einfach überfordert.

(Julian F. Reschke/wb)

PEEKs und POKEs mit List und Tücke

Wer seinem Atari etwas mehr entlocken will, als es die normalen Basic-Befehle zulassen, der sollte sich der zahlreichen Speicherstellen bedienen, die man mit PEEK und POKE erreicht.

Das Atari-Basic verfügt schon in der Grundversion über einige sehr komfortable Befehle. Wenn man jedoch beispielsweise bewegte Grafik erzeugen möchte, stößt man in Basic schnell auf Grenzen. Wer sich dann nicht recht mit Maschinensprache anfreunden will, kann sich einer Vielzahl von PEEKs und POKEs bedienen, ohne die ein gutes Programm in Basic schon fast nicht mehr auskommt.

Die Beschreibung der einzelnen Adressen geht jeweils nach einem bestimmten Schema vor sich. Zuerst findet man den dezimalen Wert der Speicherstelle vor. Danach folgt der entsprechende hexadezimale Wert, der jedoch nur bei der Programmierung in Maschinensprache interessant ist.

Viele der Player-Missile-Adressen haben die Eigenart, daß sie, je nachdem ob sie gelesen oder beschrieben werden, unterschiedliche Funktionen wahrnehmen. Bezieht sich in der Tabelle eine Beschreibung auf das Lesen, so wird dies mit (L), beim Schreiben mit (S) gekennzeichnet. Die Größe eines Players oder Missiles wird folgendermaßen angegeben. Wert 0 oder 2 entspricht der normalen Größe. Wert 1 ergibt die doppelte Größe und Wert 3 die vierfache Größe. Ist der Inhalt eines Kollisionsregisters ungleich Null, so hat ein Zusammenstoß zwischen den jeweils angegebenen Playern oder Missiles stattgefunden.

(Wolfgang Czerny/wb)

Tabelle der wichtigsten PEEKs und POKEs

Dezimal-Wert	Hexadezimal-Wert	Label	Beschreibung
16	$10	IRQEN	Die Interruptquellen des Pokey können von hier aus gesteuert werden. Beispiel: »POKE 16,64:POKE 53774,64« setzen die BREAK-Taste bis zum nächsten Graphics-Befehl außer Funktion.
17	$11	IRQST	Zeigt an, ob Interrupts des Pokey aktiv geworden sind. Beispiel: »PEEK(17)=0« sagt aus, daß die BREAK-Taste gedrückt wurde.
18,19,20	$12,$13,$14	CLOCK	Ein 3-Byte-Wert, der alle 1/50 Sekunde inkrementiert wird. Zeitdifferenzen kann man wie folgt berechnen: »Start=INT((PEEK(18)*65536)+(PEEK(19)*256)+PEEK(20))/50)«. Die STOP-Zeit wird ebenso ermittelt. Die Zeitdifferenz ZEIT errechnet sich dann aus STOP minus START in Sekunden. Eine Uhr kann mit diesen Speicherstellen natürlich auch simuliert werden.
65	$41	IOSOUNDEN	Bei einem Wert von Null wird bei Input/Output-Operationen, also beispielsweise beim Lesen einer Diskette, die Tonausgabe unterdrückt. Beispiel: »POKE 65,0« schaltet den Ton ab, »POKE 65,3« schaltet ihn wieder ein. Auf die Sound-Befehle oder das Tastaturklicken hat dies jedoch keinen Einfluß.
77	$4D	ATTRACT	Der sogenannte Attract-Modus ist für den Farbwechsel zur Schonung des Bildschirms verantwortlich. Wird für etwa zehn Minuten keine Taste betätigt, so hat ein Zähler den Wert 128 erreicht und schaltet diesen Modus ein. Dies kann verhindert werden, indem man im Programm regelmäßig den Befehl »POKE 77,0« durchführt und so den Zähler auf Null zurücksetzt. Mit »POKE 77,129« schaltet man den Attract- Modus sofort ein.
82	$52	LMARGIN	Linker Rand bei der Textdarstellung. Der Standardwert ist 2. Mit »POKE 82,0« beispielsweise kann der linke Rand auf Spalte 0 eingestellt werden.
83	$53	RMARGIN	Rechter Rand bei der Textdarstellung. Der Standardwert in Grafikstufe 0 ist 39. »POKE 83,X« stellt den rechten Rand auf Spalte X.
84	$54	ROWCRS	Diese Speicherstelle beinhaltet die aktuelle Zeilenposition des Cursors in Grafikstufe 0.
85,86	$55,$56	COLCRS	Aktuelle Spaltenposition des Cursors als 2-Byte-Wert in Grafikstufe 0.
88,89	$58,$59	SCRSTART	Dieser 2-Byte-Wert gibt die Adresse des ersten Bildschirmbytes an. Man errechnet sie folgendermaßen: »ADR=PEEK(88)+256*PEEK(89)« Kennt man diese Adresse, so kann man den Bildschirm leicht speichern oder beliebige Werte direkt auf den Bildschirm POKEn.
93	$5d	DATCURS	Hier wird das Zeichen abgelegt, das sich unmittelbar unter dem Cursor befindet.
106	$6a	RAMTOP	Durch die Verschiebung des RAM-Tops wird es beispielsweise möglich, mehrere Screens gleichzeitig im Speicher zu behalten und sie dann im sogenannten Page-Flipping hin und her zu schalten.
186,187	$ba,$bb		Dieser 2-Byte-Wert enthält die Zeilennummer beim Abbruch eines Programms. Ein solcher Abbruch kann durch eine Programmunterbrechung mit BREAK, ein STOP oder einen Fehler verursacht werden.
195	$C3	ERRSAV	Beim Auftreten eines Fehlers wird der Fehlercode hier gespeichert. Dies ist vor allem dann von Nutzen, wenn ein Programmabbruch mit dem TRAP-Befehl verhindert wird, der jeweilige Fehler aber dennoch registriert werden soll. Beispiel: »ERROR=PEEK(195)«.
559	$22f	DMACNTL	Antic-Kontrolle. Belegt man diese Speicherstelle mit dem Wert Null, so wird der Bildschirm ausgeschaltet. Je nach Grafikstufe bringt dies eine Geschwindigkeitssteigerung bis zu 30 Prozent. Um den Bildschirm wieder einzuschalten, muß man die Speicherstelle mit ihrem ursprünglichen Wert belegen. Beispiel: »WERT=PEEK(559):POKE 559,0« schaltet den Bildschirm aus. »POKE 559,WERT« schaltet ihn wieder ein. POKEt man hier den Wert 62, so bedeutet dies einfache Auflösung und beim Wert 46 doppelte Zeilenauflösung der Player und Missiles.
560,561	$230,$231	DLPTR	Dieser 2-Byte-Wert gibt die Anfangsadresse der Display List an. »DL=PEEK(560)+256*PEEK(561)«.
580	$244	COLDSTR	Wenn in dieser Speicherstelle ein Wert größer Null steht, erfolgt beim Betätigen der »SYSTEM RESET«-Taste ein Kaltstart. Das heißt, daß gewisse Initialisierungsvorgänge ablaufen und bei angeschlossenem Diskettenlaufwerk neu gebootet wird. Beispiel: »POKE 580,1«.

Dezimal-Wert	Hexa-dezimal-Wert	Label	Beschreibung
623	$26f		Prioritätsregister der Player-Missiles.
694	$2b6	XORKEYMSK	POKEt man eine Zahl in diese Speicherstelle, so wird der Zeichencode der Tastatur um den gePOKEten Wert verschoben. Beispiel: »POKE 694,128« bewirkt, daß statt der normalen Buchstaben inverse Buchstaben und Zeichen ausgegeben werden.
702	$2be	SHIFTLOCK	Hier kann man die Tastaturcodes bis zu einem gewissen Grad beeinflussen. »POKE 702,128« simuliert die Betätigung der CONTROL-Taste. Tippt man also normale Buchstaben ein, so werden statt dessen die entsprechenden Grafikzeichen ausgegeben. »POKE 702,0« schaltet den Kleinschrift-Modus ein, den man auch über die CAPS-Taste erreicht. »POKE 702,64« schaltet zurück in den normalen Schrift-Modus. »POKE 702,128« simuliert die Betätigung der CONTROL-Taste. Tippt man also normale Buchstaben ein, so werden statt dessen Grafikzeichen ausgegeben. »POKE 702,255« bewirkt, daß nur noch Tasten akzeptiert werden, die mit numerischen Werten oder Interpunktion belegt sind. Buchstaben können nur über die SHIFT-Taste erzeugt werden.
703	$2bf	NUMNXTLIN	Der hier gespeicherte Wert gibt die Zahl der Textzeilen auf dem Bildschirm an. Es werden nur die Werte 0, 4 und 24 akzeptiert. POKEt man zum Beispiel in Grafikstufe 0 den Wert 4, so erhält man ein vierzeiliges Textfenster am unteren Bildschirmrand, wie man es von verschiedenen Grafikstufen her kennt. Auf diese Weise kann man auch in Grafikstufe 0 ein vom sonstigen Bildschirm unabhängiges Textfenster einschalten. Wie in Grafikstufe 1 und 2 muß der Bildschirm dann mit PRINT #6 angesprochen werden. Dabei empfiehlt es sich, den Cursor mit »POKE 752,1« zumindest vorübergehend auszuschalten.
704	$2c0	COLPM0	Farbregister für Player 0 und Missile 0
705	$2c1	COLPM1	Farbregister für Player 1 und Missile 1
706	$2c2	COLPM2	Farbregister für Player 2 und Missile 2
707	$2c3	COLPM3	Farbregister für Player 3 und Missile 3
708	$2c4	COLOR0	Farbregister 0. Dies entspricht dem Befehl »SETCOLOR 0,Farbe,Helligkeit«. Um den entsprechenden POKE-Wert zu erhalten, multipliziert man den Farbwert mit 16 und addiert den Helligkeitswert. Beispiel für die Farbe Blau. »SETCOLOR 0,9,4« entspricht dem Befehl »POKE 708,9*16+4« oder »POKE 708,148«. In Grafikstufe 1 und 2 bestimmt dieses Register die Farbe der Großbuchstaben.
709	$2c5	COLOR1	Farbregister 1. In Grafikstufe 1 und 2 bestimmt dieses Register die Farbe von Kleinbuchstaben. In Stufe 0 und 8 wird hier die Helligkeit der Zeichen eingestellt.
710	$2c6	COLOR2	Farbregister 2. In Grafikstufe 1 und 2 bestimmt dieses Register die Farbe inverser Großbuchstaben, in Stufe 0 und 8 die Farbe des Hintergrunds.
711	$2c7	COLOR3	Farbregister 3. Werden die vier Missiles zu einem Player vereinigt, so wird die Farbe dieses zusätzlichen Players hier festgelegt. In Grafikstufe 1 und 2 steht hier der Farbwert für inverse Kleinbuchstaben.
712	$2c8	COLOR4	Hintergrundfarbe
729	$2d9	KEYRPDELY	Bestimmt die Zeit vom Tastendruck bis zum Beginn der Repeat-Funktion. Der Standardwert ist 40.
730	$2da	KEYREP	Gibt die Wiederholungsfrequenz der Repeat-Funktion an. Der Standardwert ist 5.
731	$2db	CLICKDISA	Hier kann man das Tastaturklicken ein- und ausschalten. »POKE 731,255« bedeutet Klicken aus, »POKE 731,0« schaltet das Klicken ein.
732	$2dc	HELPFLAG	Diese Speicherstelle gibt an, ob die HELP-Taste gedrückt wurde. Beispiel: »X=PEEK(732)« x= 17 bedeutet HELP-Taste gedrückt x= 81 bedeutet SHIFT-HELP gedrückt x=145 bedeutet CONTROL-HELP gedrückt
752	$2f0	CRSINH	Mit dieser Speicherstelle läßt sich der Cursor ein- und ausstellen. »POKE 752,1« bedeutet Cursor aus, »POKE 752,0« bedeutet Cursor ein.
755	$2f3	CHARCTRL	Hier kann man die Darstellung des Cursors und des Textes auf dem Bildschirm beeinflussen. Es sind acht verschiedene Einstellungen möglich. Beispiel: »POKE 755,X« X=0 - Zeichen sichtbar, Cursor unsichtbar, Schrift normal X=1 - Zeichen unsichtbar, Cursor unsichtbar, Schrift normal X=2 - Zeichen sichtbar, Cursor sichtbar, Schrift normal X=3 - Zeichen unsichtbar, Cursor sichtbar, Schrift normal X=4 - Zeichen sichtbar, Cursor unsichtbar, Schrift auf dem Kopf X=5 - Zeichen unsichtbar, Cursor unsichtbar, Schrift auf dem Kopf X=6 - Zeichen sichtbar, Cursor sichtbar, Schrift auf dem Kopf X=7 - Zeichen unsichtbar, Cursor sichtbar, Schrift auf dem Kopf POKEt man mit einer Schleife abwechselnd die Werte 0 bis 3 in diese Speicherstelle, so kann man invers dargestellten Text in Grafikstufe 0 zum Blinken bringen.
756	$2f4	CHARBASE	Zeichensatzauswahl. Der Atari hat neben seinem Standardzeichensatz noch einen internationalen Zeichensatz vorrätig, in dem auch die deutschen Umlaute enthalten sind. Dieser internationale Zeichensatz wird durch »POKE 756,204« aktiviert. Mit »POKE 756,224« schaltet man zurück in den normalen Zeichensatz. Der »POKE 756,226« ermöglicht es, in Grafikstufe 1 und 2 Kleinbuchstaben und Grafikzeichen darzustellen. Allerdings muß man sich dann das Leerzeichen selbst definieren, da in diesem Modus das Leerzeichen als Herz dargestellt wird.

Dezimal-Wert	Hexa-dezimal-Wert	Label	Beschreibung
764	$2fc	KBCODE	Hier wird der Tastaturcode der zuletzt gedrückten Taste gespeichert. Dieser Code entspricht allerdings nicht dem Standard-ASCII-Code, sondern stellt einen Atari-spezifischen Code dar.
1021-1151	$3fd–$47f	CASBUF	Kassettenpufferbereich. Bei der Verwendung von Diskettenlaufwerken kann man diesen Bereich zum Speichern von Maschinenunterprogrammen verwenden.
1536-1791	$0600–06ff		Page 6 – Dieser Bereich wird in der Regel zum Speichern von Maschinenunterprogrammen verwendet.
53248	$d000		(L) Kollisionsregister Missile 0 mit Spielfeld. (S) Horizontale Position von Player 0.
53249	$d001		(L) Kollisionsregister Missile 1 mit Spielfeld. (S) Horizontale Position von Player 1.
53250	$d002		(L) Kollisionsregister Missile 2 mit Spielfeld. (S) Horizontale Position von Player 2.
53251	$d003		(L) Kollisionsregister Missile 3 mit Spielfeld. (S) Horizontale Position von Player 3.
53252	$d004		(L) Kollisionsregister Player 0 mit Spielfeld. (S) Horizontale Position von Missile 0.
53253	$d005		(L) Kollisionsregister Player 1 mit Spielfeld. (S) Horizontale Position von Missile 1.
53254	$d006		(L) Kollisionsregister Player 2 mit Spielfeld. (S) Horizontale Position von Missile 2.
53255	$d007		(L) Kollisionsregister Player 3 mit Spielfeld. (S) Horizontale Position von Missile 3.
53256	$d008		(L) Kollisionsregister Missile 0 mit einem Player. (S) Größe von Player 0.
53257	$d009		(L) Kollisionsregister Missile 1 mit einem Player. (S) Größe von Player 1.
53258	$d00a		(L) Kollisionsregister Missile 2 mit einem Player. (S) Größe von Player 2.
53259	$d00b		(L) Kollisionsregister Missile 3 mit einem Player. (S) Größe von Player 3.
53260	$d00c		(L) Kollisionsregister Player 0 mit anderem Player. (S) Größe von Missile 0.
53261	$d00d		(L) Kollisionsregister Player 1 mit anderem Player. (S) Größe von Missile 1.
53262	$d00e		(L) Kollisionsregister Player 2 mit anderem Player. (S) Größe von Missile 2.
53263	$d00f		(L) Kollisionsregister Player 3 mit anderem Player. (S) Größe von Missile 3.
53277	$d01d		(S) Hier kann man die Player und Missile ein- und ausschalten.
53278	$d01e		(S) Schreibt man in dieses Register einen beliebigen Wert, so werden alle Kollisionsregister gelöscht.
53279	$d01f	CONSOL	Diese Speicherstelle wird zur Abfrage der Sondertasten des Atari, also von OPTION, SELECT und START verwendet. Da die HELP-Taste nur bei den XL-Modellen vorkommt, ist ihre Abfrage an anderer Stelle zu entnehmen. Die Werte, die man aus dieser Speicherstelle erhält, haben folgende Bedeutung: X=PEEK(53279) x=0 – OPTION+SELECT+START gedrückt x=1 – OPTION+SELECT gedrückt x=2 – OPTION+START gedrückt x=3 – OPTION Taste gedrückt x=4 – SELECT+START gedrückt x=5 – SELECT Taste gedrückt x=6 – START Taste gedrückt x=7 – keine der Sondertasten wurde betätigt Die gleichzeitige Betätigung von CONTROL oder SHIFT mit einer der Sondertasten liefert keine anderen Werte.
53770	$d20a	RANDOM	Diesem Register kann man eine Zufallszahl entnehmen, deren Wert zwischen 0 und 255 liegt. Das ist vor allem beim Arbeiten mit Compilern wichtig, die die »RND«-Funktion nicht übersetzen.
54016	$d300	PORTA	Port A der PIA
54017	$d301	PORTB	Port B der PIA
54018	$d302	PORTACNTL	Datenrichtungsregister für Port A.
54019	$d303	PORTBCNTL	Datenrichtungsregister für Port B.
54279	$d407	PMBASE	Hier muß das High-Byte der Adresse des Player-Missile-Speichers stehen.

Dem SOUND-Befehl auf der Spur

Oft wird den Atari-Computern Ihre Soundfähigkeit abgesprochen. Ein paar Tricks entlocken dem Atari aber dennoch sehr wohlklingende Töne.

PROGRAMM-STECKBRIEF	
Programmname	Musik-Kurs
Programmtyp	Kurs
Programmiersprache	Atari-Basic
Programmlänge	insgesamt 33716 Byte
für Computer	800XL/130XE
zusätzliche Hardware	Diskettenlaufwerk oder Kassettenrecorder
Eingabehilfe	Prüfsummer
Bemerkung	zusammenhängender Musik-Kurs (SOUND.7 und SOUND.8 sind Demos).
Leserservice	Diskette (SOUND.12 bis SOUND.8)

Es ist sicher etwas zuviel verlangt, einem Heimcomputer Musikqualitäten einer elektronischen Orgel abzuverlangen. Schließlich wurde er in erster Linie für andere Zwecke entwickelt. So geht seine Verwendung vom Spielecomputer über Grafikanwendungen bis hin zur Dateiverwaltung und Textverarbeitung. Aber es beschäftigen sich eben auch einige »Musikfreaks« mit den Soundeffekten.

In diesem Beitrag dreht sich alles um den SOUND-Befehl. So gilt es hier, dem Atari-Computer in Sachen Tonerzeugung auf die Sprünge zu helfen. Der eingebaute Sound-Baustein »Pokey« soll also voll ausgereizt werden. Dabei werden Tricks angewandt, die bislang nur Insidern bekannt waren.

Noch ein Wort zum Aufbau dieses Beitrags. Sie finden eine ganze Reihe von dokumentierten Listings, die das Grundgerüst des Artikels darstellen. Nur wenn Sie diese Programme vorliegen haben, können Sie sich richtig in die »Sound«-Materie der Atari-Computer einarbeiten. Es empfiehlt sich also, alle hier abgedruckten Programme einzugeben. Wenn Sie ein Diskettenlaufwerk verwenden, sollten Sie die Programme mit den vorgeschlagenen Programmnamen speichern. Nach der Abarbeitung des ersten Programms wird dann automatisch das nächste Programm geladen etc. Verwenden Sie einen Kassettenrecorder, müssen Sie die Programme in der abgedruckten Reihenfolge eingeben und auf einer Kassette speichern. Übrigens: Lassen Sie die Musikprogramme nur unter dem normalen Atari-Basic laufen, Turbo-Basic XL arbeitet die Programme viel zu schnell ab.

Der SOUND-Befehl setzt sich aus den vier Parametern »Voice«, »Pitch«, »Distortion« und »Volume« zusammen. Dabei steht Voice für Stimme, Pitch für Tonhöhe, Distortion für Verzerrung und Volume für die Lautstärke. Damit man sich besser und schneller in die Beispielsprogramme einarbeiten kann, werden immer die gleichen Variablen verwendet. Dies trägt außerdem zur Übersichtlichkeit bei. In den folgenden Beispielen wird immer nur die erste Stimme verwendet. Der Verzerrungsfaktor beträgt dabei stets 10. Die Tonwiedergabe erfolgt bei dieser Einstellung verzerrungsfrei.

Falls Sie ein Diskettenlaufwerk verwenden, geben Sie jetzt bitte »RUN "D:SOUND.12« ein, um Listing 1 zu starten. Beim Kassettenrecorder laden Sie bitte das entsprechende Programm.

Innerhalb der Beispielsprogramme können Sie sich mit der OPTION-Taste eine Tonfolge immer wieder anhören. Mit START gelangen Sie zum nächsten Beispiel.

Der Ablauf von Listing 1:
- Der Computer meldet sich zuerst mit einem mittleren C.
- Verändern der Lautstärke, mal langsam und dann wieder schnell.
- nur die Tonhöhe verändern
- Tonhöhe und Lautstärke verändern

Wenden wir uns nun der Musikprogrammierung zu. Insgesamt stehen Pitch-Werte von 0 bis 255 zur Verfügung. Es sind aber nur die Werte interessant, die jeweils einem Ton entsprechen (siehe Tabelle 1).

Notennummer	Tonhöhe	Notenname	Notennummer	Tonhöhe	Notenname
1	14	C	26	64	B
2	55	B	27	68	A# o. Bb
3	16	A# o. Bb	28	72	A
4	17	A	29	76	G# o. Ab
5	18	G# o. Ab	30	81	G
6	19	G	31	85	F# o. Gb
7	21	F# o. Gb	32	91	F
8	22	F	33	96	E
9	23	E	34	102	D# o. Eb
10	24	D# o. Eb	35	108	D
11	26	D	36	114	C# o. Db
12	27	C# o. Db	37	121	C
13	29	C	38	128	B
14	31	B	39	136	A# o. Bb
15	33	A# o. Bb	40	144	A
16	35	A	41	153	G# o. Ab
17	37	G# o. Ab	42	162	G
18	40	G	43	173	F# o. Gb
19	42	F# o. Gb	44	182	F
20	45	F	45	193	E
21	47	E	46	204	D# o. Eb
22	50	D# o. Eb	47	217	D
23	53	D	48	230	C# o. Db
24	57	C# o. Db	49	243	C
25	60	C	50	255	B

Tabelle 1. Tonhöhentabelle

Ein Unterprogramm bestimmt die Tondauer, das heißt, ob zum Beispiel eine ganze, halbe oder Viertelnote gespielt wird. Es bietet sich hier an, einen im Betriebssystem der Atari-Computer eingebauten Counter (Zähler) zu verwenden. Dazu eignet sich die Speicherstelle 540. Der in dieser Adresse enthaltene Zähler zählt jede 1/50-Sekunde einen Schritt zurück. In die Adresse 540 kann also ein Wert zwischen 0 und 255 gePOKEt werden. Nach Beendigung der Warteschleife, also wenn die Adresse den Wert 0 erreicht hat, kann ins Hauptprogramm zurückgesprungen werden. Hier eine Routine, die eine Sekunde lang eine Note spielt:

```
10 SOUND 1,200,10,10:REM
```
Tonausgabe einschalten
```
20 POKE 540,50:REM
```
Der Ton soll für eine Sekunde gespielt werden
```
30 IF PEEK(540)< > 0 THEN 30:REM
```
Wiederhole bis die Speicherstelle 540 den Wert 0 enthält
```
40 SOUND 1,0,0,0:REM
```
Ton abstellen.

Da in einem Programm eine solche Warteschleife öfter benötigt wird, bietet es sich natürlich an, für diesen Zweck eine separate Unterroutine zu verwenden. Weiterhin emp-

fiehlt sich eine Übergabevariable einzusetzen, die bestimmt, wie lange ein Ton gespielt werden soll.

Nachdem eine bestimmte Tonsequenz abgearbeitet wurde, müssen gegebenenfalls alle vier Stimmen wieder abgestellt werden. Die folgende Programmzeile erledigt dies für Sie:

```
60 FOR AUS=0 TO 3:SOUND AUS,0,0,0:NEXT AUS:RETURN
```

Wenn man jetzt also ein Musikstück in Basic schreiben möchte, kann man die benötigten Notennummern in DATA-Zeilen ablegen. Mit einer READ-DATA-Schleife lassen sich dann die Werte lesen und verarbeiten. Listing 2 demonstriert dieses Verfahren.

Wenn Sie ein Diskettenlaufwerk verwenden, wird Listing 2 automatisch geladen und gestartet. Besitzer eines Kassettenrecorders müssen jetzt wieder das entsprechende Programm laden.

Begleitung mit Dur-Akkorden

Haben Sie bislang nur einfache Beispiele gehört, folgt jetzt mit Listing 3 eine komplexere Melodie.

Es beginnt mit der Dimensionierung der Variablen. Hier werden einerseits LINE$ und ein Array N sowie die Variablen V0 bis V3 vordefiniert. In der Variablen N sind dann insgesamt 50 Tonhöhen gespeichert. LINE$ nimmt immer eine Textzeile auf und stellt sie auf dem Bildschirm dar. Sie dient sozusagen dem Dialog. Anschließend erfolgt ein Sprung zu Zeile 100. Die DATA-Werte ab Zeile 100 werden aber zunächst ignoriert, und es wird ein Unterprogramm ab Zeile 21000 aufgerufen.

Dieses Unterprogramm ist für das Löschen des Bildschirms, die Hintergrundfarbe, die Schrift und die Fixierung der Bildränder zuständig. Dazu werden folgende POKEs verwendet:

- »POKE 752,1« Cursor aus
- »POKE 82,X« und »POKE 83,X« für das Setzen des linken und rechten Bildschirmrandes
- »POKE 201,X« setzt die Anzahl der Leerzeichen fest, die bei »PRINT,« ausgegeben werden sollen
- »POKE 77,0« schaltet den automatischen Farbwechsel aus.

Nachdem das Unterprogramm abgearbeitet ist, springt das Programm zurück zu Zeile 120. Jetzt werden noch die Noten in das Feld N geschrieben.

Eigentlich stellt sich nun die Frage: Warum werden die Tonhöhen-Werte noch zusätzlich in einem Feld abgelegt? Es würde doch genügen, die benötigten Werte direkt aus den DATA-Zeilen zu lesen. Das ist prinzipiell richtig, aber wenn Sie Tabelle 1 von oben nach unten lesen, wird Ihnen auffallen, daß in der mittleren Spalte verschiede Werte fehlen. In der ersten Spalte, mit der Bezeichnung »Noten-Nummer«, finden Sie aber 50 fortlaufende Nummern. Diese Nummern erlauben eine schnellere Berechnung der Akkorde. Dazu muß nur der Grundton des gewünschten Akkordes angegeben werden und schon wird der gesamte Akkord ausgegeben.

Die Unterroutine zur Berechnung der Akkorde beginnt ab Zeile 40 und nennt sich »Akkord-Berechner«. Vor dem Aufruf dieser Routine muß die Notennummer in der Variablen P übergeben werden. Dabei darf P nur Werte zwischen 8 und 50 enthalten. Weiterhin nimmt noch die Variable P0 die entsprechenden Tonhöhen aus dem Feld N an. Dann wird der Akkord berechnet und die Tonhöhenwerte in den Variablen P1, P2 und P3 abgelegt. In Zeile 42 werden die vier Stimmen aktiviert. Stimme 0 ist dabei auf eine höhere Lautstärke gesetzt als die restlichen drei Stimmen des Akkords.

Das Unterprogramm ab Zeile 30 ändert lediglich die Tonhöhe und springt daraufhin in die Verzögerungsschleife. In Zeile 70 sind alle vier Stimmen zunächst auf gleiche Lautstärke gesetzt. Diese wird dann gleichmäßig zurückgenommen, bis nichts mehr zu hören ist.

Zeile 210 liest nun die Daten für LINE$, CHORD, P und WAIT. Der Inhalt von LINE$ wird auf dem Bildschirm ausgegeben und anschließend das Unterprogramm ab Zeile 40 aufgerufen. In dieser Routine steht also in LINE$ »Häns-chen klein – ging allein«, die Variable CHORD enthält nun den Wert 49, P den Wert 30 und WAIT den Wert 30.

Der in Listing 3 verwendete Algorithmus ist nicht für jedes Lied geeignet. Eventuell müssen Sie einige Änderungen vornehmen. Schließlich gibt es verschiedene Tempi und nicht nur Dur-Akkorde. Das abgedruckte Programm zeigt ein Beispiel, wie man die entsprechenden Noten für Akkorde findet und so ein Lied komponiert.

Laden Sie jetzt Listing 4. Dieses Programm spielt jeweils eine Tonleiter in Dur, Moll, und C-Dur vor. Begleitend erscheinen die entsprechenden Listing-Zeilen auf dem Bildschirm. So können Sie sich auch ein Bild von der Programmiertechnik machen.

GePOKEte Musik

Mit den ersten sechs Programmen (Listing 1 bis Listing 6) hatten Sie die Gelegenheit, sich einen Eindruck davon zu verschaffen, was der SOUND-Befehl zu leisten vermag. Allerdings gibt es einen noch viel besseren Weg, nämlich von Maschinensprache aus. Dann kann man die Fähigkeiten des Pokey-Bausteins, der für die Tonausgabe zuständig ist, voll ausnutzen; denn für manche Geräuscheffekte ist das Atari-Basic einfach zu langsam. Sie müssen aber nicht unbedingt auf Maschinensprache zurückgreifen, denn auch der POKE-Befehl kann die Sound-Register direkt ansprechen. Auf diese Art und Weise lassen sich dann noch viel wirkungsvollere Klangvariationen erzeugen. Listing 7 stellt ein solches Klangbeispiel vor. Zur Bedienung brauchen Sie lediglich einen Joystick. Schließen Sie diesen bitte an Joystickport 1 an.

Auf dem Bildschirm erscheinen dann die Zahlen 53760 bis 53768. Sie repräsentieren die Speicherstellen, die für die Tonausgabe zuständig sind. Nach dem Starten des Programms sind alle Adressen auf Null gesetzt. In der ersten Zeile befindet sich noch ein ">"-Zeichen. Bewegen Sie es mit dem Joystick bis zur Zahl 53768, indem Sie den Joystick nach unten auslenken. Mit dem Feuerknopf verändern Sie die Speicherstelle. Dann brauchen Sie den Joystick nur nach vorne oder hinten zu bewegen. Hat die Adresse den gewünschten Wert erreicht, betätigen Sie nochmals die Feuertaste, und das ">"-Zeichen kehrt zur ersten Zeile

Variable	Bedeutung
V0	Stimme 0
V1	Stimme 1
V2	Stimme 2
V3	Stimme 3
P	Tonhöhe
D	Verzerrung
V	Lautstärke
Es gibt insgesamt vier Stimmen (V0 bis V3). Diesen Variablen wird stets ein Wert zugewiesen, der sich dann während des Programmlaufs nicht mehr ändert. Sie erhalten folgende Werte:	
V0	0
V1	1
V2	2
V3	3
P	Tonhöhe, kann einen ganzzahligen Wert zwischen 0 und 255 annehmen
D	Verzerrung, nimmt Werte zwischen 0 und 14 an
V	Lautstärke, für diese Variablen sind Werte zwischen 0 und 15 zugelassen

Variablenliste



zurück. »POKEn« Sie beispielsweise in die Adresse 53768 den Wert 80, in Adresse 53760 den Wert 10 und in 53761 einen beliebigen Wert. Anschließend wird der Lautsprecher Ihres Fernsehers oder Monitors aktiv.

Die Speicherzelle 53768 ist der Schlüssel zu allem. Belegen Sie diese Speicherzelle auch einmal mit anderen Werten zwischen 0 und 255. Ändern Sie dann noch die Speicherzellen 53760 und 53761. Was ist nun mit 53762 bis einschließlich 53767? Ändern Sie jetzt auch diese Speicherzellen. So können Sie sich auf experimentellem Wege einige gut klingende Geräusche aussuchen.

Bevor Sie die POKE-Befehle in Basic anwenden, müssen Sie wissen, daß der Sound-Prozessor zunächst installiert werden muß. Dazu reicht es aus, am Anfang eines Programms den Befehl »SOUND 0,0,0,0« auszuführen. Weiterhin sind die Sound-Register des Atari-Computers nur für Zahlenwerte empfänglich. POKEt man beispielsweise in die Adresse 53762 den Wert 10, erhält man nach »PEEK(53762)« nicht mehr denselben Wert. Warum dies so ist, ist für die Tonerzeugung nicht wichtig. Man muß nur wissen, daß man mit dem POKE-Befehl die Geräuscherzeugung beeinflussen kann. Tabelle 2 listet noch die englischen Bezeichnungen für die Sound-Register auf.

Falls Sie ein Diskettenlaufwerk verwenden, wird am Ende automatisch Listing 8 geladen. Es demonstriert, daß der Atari-Computer schon von Basic aus sehr wohlklingende Töne von sich zu geben im Stande ist. Auch Sie können solche Melodien komponieren, vorausgesetzt Sie haben die hier erläuterten Grundlagen verstanden. Aber auch in der Musikprogrammierung gilt das Motto: Übung macht den Meister.

Möchten Sie sich am Schluß noch eine wirklich meisterhaft umgesetzte Melodie anhören? Dann geben Sie noch Listing 9 ein. Nachdem Sie das Programm gestartet haben, hören Sie die Melodie »Digi Loo, Digi Lee«. Das Programm wurde uns freundlicherweise von Kemal Ezcan zur Verfügung gestellt; er ist einer der wenigen, die die Soundprogrammierung des Atari wirklich perfekt beherrschen. Vielleicht machen Sie ihm bald Konkurrenz? (Peter Gerstner/wb)

Register	Bezeichnung	
53760	Voice 0	Frequency
53761	Voice 0	Control
53762	Voice 1	Frequency
53763	Voice 1	Control
53764	Voice 2	Frequency
53765	Voice 2	Control
53766	Voice 3	Frequency
53767	Voice 3	Control
53768	Audio	Control

Tabelle 2. Die Sound-Register

```
0 REM SOUND.1      Einfuehrung
(c) 1985 by Peter Gerstner 08/10/85      <SA>
1 REM                                    <UQ>
2 REM                                    <UR>
3 REM                                    <US>
10 V0=0:V1=1:V2=2:V3=3:DIM TYPE$(40),N(5
0):GOTO 100                              <ED>
50 POKE 540,WAIT                         <VY>
52 IF PEEK(540)<>0 THEN 52               <OU>
54 RETURN                                <ML>
60 FOR OFF=0 TO 3:SOUND OFF,0,0,0:NEXT O
FF:RETURN                                <KN>
70 LINE=LEN(TYPE$):POSITION HOR,VER      <NG>
72 FOR ME=1 TO LINE:? TYPE$(ME,ME);:IF T
YPE$(ME,ME)="▲" THEN 76                  <DL>
74 SOUND 0,25,4,6:FOR DECAY=6 TO 0 STEP
-0.5:SOUND 0,10,0,DECAY:NEXT DECAY       <AX>
76 NEXT ME:RETURN                        <NZ>
100 GOSUB 21000:WAIT=60                  <VW>
200 TYPE$="Bitte▲den▲Begleittext▲fuer▲SO
UND.12":HOR=3:VER=9:GOSUB 70             <DL>
210 TYPE$="lesen▲und▲erst▲die▲START-Tast
e▲druecken":HOR=1:VER=11:GOSUB 70        <QS>
220 TYPE$="wenn▲es▲im▲Text▲verlangt▲wird
.":HOR=5:VER=13:GOSUB 70                 <TV>
230 GOSUB 20000:GOSUB 21000              <XK>
700 TYPE$="SOUND▲Befehl▲fuer▲das▲mittler
e▲'C'.▲":HOR=3:VER=7:GOSUB 70:GOSUB 50   <OG>
720 POSITION 12,13:? "SOUND▲0,121,10,8":
SOUND 0,121,10,8:GOSUB 50:GOSUB 60       <IG>
740 LINE=720:GOSUB 22000                 <UQ>
780 TYPE$="SOUND▲▲mit▲variabler▲LAUTSTAE
RKE":HOR=4:VER=7:GOSUB 70                <MO>
820 POSITION 12,13:? "SOUND▲0,60,10,0"   <AZ>
840 FOR D=0 TO 14:POSITION 12,13:? "SOUN
D▲0,121,10,";D;"▲":SOUND V0,121,10,D     <NM>
860 WAIT=20:GOSUB 50:NEXT D              <DD>
880 FOR D=13 TO 0 STEP -1:POSITION 12,13
:? "SOUND▲0,121,10,";D;"▲":SOUND V0,121,
10,D                                     <LV>
900 WAIT=20:GOSUB 50:NEXT D:WAIT=60:GOSU
B 50                                     <JA>
920 FOR D=0 TO 14:POSITION 12,13:? "SOUN
D▲0,121,10,";D;"▲":SOUND V0,121,10,D     <NJ>
940 WAIT=10:GOSUB 50:NEXT D              <CQ>
960 FOR D=13 TO 0 STEP -1:POSITION 12,13
:? "SOUND▲0,121,10,";D;"▲":SOUND V0,121,
10,D                                     <LS>
980 WAIT=10:GOSUB 50:NEXT D:WAIT=60:GOSU
B 50                                     <JG>
1000 FOR D=0 TO 14:POSITION 12,13:? "SOU
ND▲0,121,10,";D;"▲":SOUND V0,121,10,D:NE
XT D                                     <BI>
1020 FOR D=12 TO 0 STEP -1:POSITION 12,1
3:? "SOUND▲0,121,10,";D;"▲":SOUND V0,121
,10,D:NEXT D                             <IB>
1030 WAIT=60:GOSUB 50                    <ER>
1040 FOR D=0 TO 14 STEP 2:POSITION 12,13
:? "SOUND▲0,121,10,";D;"▲":SOUND V0,121,
10,D:NEXT D                              <YI>
1060 FOR D=12 TO 0 STEP -2:POSITION 12,1
3:? "SOUND▲0,121,10,";D;"▲":SOUND V0,121
,10,D:NEXT D:GOSUB 50                    <OV>
1080 FOR D=14 TO 0 STEP -2:POSITION 12,1
3:? "SOUND▲0,121,10,";D;"▲":SOUND V0,121
,10,D:NEXT D:GOSUB 50                    <QB>
1100 FOR D=14 TO 0 STEP -1:POSITION 12,1
3:? "SOUND▲0,121,10,";D;"▲":SOUND V0,121
,10,D:NEXT D:GOSUB 50                    <OF>
1120 FOR D=14 TO 0 STEP -0.5:POSITION 12
,13:? "SOUND▲0,121,10,";INT(D);"▲":SOUND
 V0,121,10,D:NEXT D:GOSUB 50             <CV>
1140 FOR D=14 TO 0 STEP -0.2:POSITION 12
,13:? "SOUND▲0,121,10,";INT(D);"▲":SOUND
 V0,121,10,D:NEXT D:GOSUB 50             <ZV>
1160 POSITION 12,13:? "SOUND▲0,121,10,10
":SOUND 0,121,10,10:WAIT=20:GOSUB 50     <VM>
1180 POSITION 12,13:? "SOUND▲0,121,10,0▲
":GOSUB 60:WAIT=20:GOSUB 50              <JV>
1200 FOR ME=1 TO 3:POSITION 12,13:? "SOU
ND▲0,121,10,10":SOUND 0,121,10,10:WAIT=1
0:GOSUB 50                               <NP>
1220 POSITION 12,13:? "SOUND▲0,121,10,0▲
":GOSUB 60:GOSUB 50:NEXT ME:WAIT=60:GOSU
B 50                                     <BP>
1240 FOR ME=1 TO 5:POSITION 12,13:? "SOU
ND▲0,121,10,10":SOUND 0,121,10,10        <TN>
1260 POSITION 12,13:? "SOUND▲0,121,10,0▲
":GOSUB 60:NEXT ME:WAIT=60:GOSUB 50      <MZ>
1300 LINE=820:GOSUB 22000                <YP>
1360 TYPE$="SOUND▲mit▲variabler▲TONHOEHE
":HOR=6:VER=7:GOSUB 70                   <MR>
1400 FOR P=0 TO 255:POSITION 12,13:? "SO
UND▲0,";P;",10,8▲▲":SOUND 0,P,10,8:NEXT
P                                        <OZ>
1420 FOR P=255 TO 0 STEP -1:POSITION 12,
13:? "SOUND▲0,";P;",10,8▲▲":SOUND 0,P,10
```

Listing 1. Einfache Beispiele mit dem SOUND-Befehl. Bitte mit dem Namen »D:SOUND.12« auf Diskette speichern.

```
,8:NEXT P                                                <EU>
1440 FOR P=0 TO 255 STEP 5:POSITION 12,1
3:? "SOUND▴0,";P;",10,8▴▴":SOUND 0,P,10,
8:NEXT P                                                 <DD>
1460 FOR P=255 TO 0 STEP -5:POSITION 12,
13:? "SOUND▴0,";P;",10,8▴▴":SOUND 0,P,10
,8:NEXT P                                                <JK>
1480 FOR P=0 TO 250 STEP 10:POSITION 12,
13:? "SOUND▴0,";P;",10,8▴▴":SOUND 0,P,10
,8:NEXT P                                                <EK>
1500 FOR P=250 TO 0 STEP -10:POSITION 12
,13:? "SOUND▴0,";P;",10,8▴▴":SOUND 0,P,1
0,8:NEXT P                                               <MG>
1520 POSITION 12,13:? "SOUND▴0,0,0,0▴▴▴"
:GOSUB 60:LINE=1400:GOSUB 22000                          <CB>
1560 TYPE$="SOUND▴▴mit▴variabler":HOR=10
:VER=5:GOSUB 70                                          <KT>
1580 TYPE$="TONHOEHE▴und▴LAUTSTAERKE":HO
R=8:VER=7:GOSUB 70:WAIT=60:GOSUB 50:REST
ORE 1600                                                 <UX>
1600 DATA▴60,53,47,45,40,35,31,29,29,31,
35,40,45,47,53,60                                        <EE>
1620 FOR ME=1 TO 16:READ P:FOR V=14 TO 0
 STEP -2:POSITION 12,13:? "SOUND▴0,";P;"
,10,";V;"▴":SOUND 0,P,10,V                               <AS>
1640 NEXT V:WAIT=10:GOSUB 50:NEXT ME:RES
TORE 1660                                                <YR>
1660 DATA▴243,193,162,121,96,81,60,47,40
,29,23,19,14,14,19,23,29,40,47,60,81,96,
121,162,193,243                                          <BR>
1680 FOR ME=1 TO 26:READ P:FOR V=14 TO 0
 STEP -2:POSITION 12,13:? "SOUND▴0,";P;"
,10,";V;"▴":SOUND 0,P,10,V                               <CC>
1690 NEXT V:WAIT=10:GOSUB 50:NEXT ME                     <BJ>
1700 RESTORE 1600:LINE=1620:GOSUB 22000                  <NX>
2000 REM                                                 <SZ>
2001 REM                                                 <TD>
2002 REM SOUND.2   BASIC-MUSIK                           <XB>
2003 REM                                                 <TL>
2004 REM                                                 <TP>
2050 RESTORE 2100                                        <IL>
2100 DATA▴14,15,16,17,18,19,21,22,23,24,
26,27,29,31,33,35,37,40,42,45,47,50,53,5
7,60,64,68,72,76,81,85,91,96                             <WE>
2110 DATA▴102,108,114,121,128,136,144,15
3,162,173,182,193,204,217,230,243,255                    <XI>
2120 GOSUB 21100:FOR X=1 TO 50:READ IT:N
(X)=IT:NEXT X                                            <XW>
2130 TYPE$="Lesen▴Sie▴bitte▴im▴Text▴weit
er":HOR=5:VER=9:GOSUB 70                                 <GG>
2140 TYPE$="druecken▴Sie▴erst▴die▴START-
Taste,":HOR=3:VER=11:GOSUB 70                            <YO>
2150 TYPE$="wenn▴es▴im▴Text▴verlangt▴wir
d.":HOR=5:VER=13:GOSUB 70                                <CT>
2200 GOSUB 20000:GOSUB 21100                             <HV>
2300 LIST 2310                                           <QC>
2310 FOR P=1 TO 50:FOR V=15 TO 0 STEP -3
:SOUND 0,N(P),10,V:NEXT V:NEXT P                         <EN>
2320 IF LINE<>2310 THEN ? :LIST 2330                     <QO>
2330 FOR P=50 TO 1 STEP -1:FOR V=15 TO 0
 STEP -3:SOUND 0,N(P),10,V:NEXT V:NEXT P                 <YM>
2335 IF LINE=2310 THEN 2400                              <QE>
2400 LINE=2310:GOSUB 22000                               <YP>
2500 RUN "D:SOUND.3":REM CASSETTEN BENUE
TZER MUESSEN HIER GRAPHICS 0:END ANSTATT
 RUN"D:SOUND.3" EINGEBEN                                 <BX>
20000 POSITION 2,20:POKE 201,4:POKE 752,
1                                                        <PR>
20010 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL E}"                                           <AT>
20020 ? ,"{SHIFT =}▴Druecke▴START▴fuer▴w
eiter▴{SHIFT =}"                                         <PW>
20030 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL C}"                                           <BW>
20040 IF PEEK(53279)=6 THEN POKE 755,2:R
ETURN                                                    <BQ>
20050 POKE 755,3:POKE 755,2:PET=0:GOTO 2
0040                                                     <NR>
21000 GRAPHICS 0:SETCOLOR 2,9,0:SETCOLOR
 4,9,0:SETCOLOR 1,9,12:POKE 752,1:POKE 8
2,2:POKE 83,39:POKE 201,8                                <IN>
21010 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL E}"                          <JM>
21020 ? ,"{SHIFT =}▴Der▴SOUND▴Befehl▴{SH
IFT =}"                                                  <IV>
21030 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL C}"                          <LH>
21040 POKE 77,0:RETURN                                   <AK>
21100 GRAPHICS 0:SETCOLOR 2,9,0:SETCOLOR
 4,9,0:SETCOLOR 1,9,12:POKE 752,1:POKE 8
2,2:POKE 83,39:POKE 201,11                               <YD>
21110 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL E}"              <MQ>
21120 ? ,"{SHIFT =}▴BASIC▴MUSIK▴{SHIFT =
}"                                                       <IJ>
21130 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL C}"                     <OV>
21140 POKE 77,0:RETURN                                   <AN>
22000 POSITION 2,19:POKE 201,5:POKE 752,
1                                                        <WW>
22010 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL E}" <XZ>
22015 ? ,"{SHIFT =}▴OPTION▴=▴▴Wiederholu
ng▴▴{SHIFT =}"                                           <UV>
22020 ? ,"{SHIFT =}▴START▴▴=▴▴Weiter▴▴▴▴
▴▴▴▴{SHIFT =}"                                           <JE>
22030 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL C}" <ZG>
22040 IF PEEK(53279)=6 THEN POKE 755,2:G
OSUB 21000:RETURN                                        <TV>
22050 IF PEEK(53279)=3 THEN POP :POKE 75
5,2:POSITION 2,19:? "{ESC SHIFT DEL}{ESC
 SHIFT DEL}{ESC SHIFT DEL}{ESC SHIFT DEL
}";:GOTO LINE                                            <OW>
22060 POKE 755,3:POKE 755,2:GOTO 22040                   <XL>
25000 POSITION 2,15:? "{ESC SHIFT DEL}{E
SC SHIFT DEL}{ESC SHIFT DEL}{ESC SHIFT D
EL}{ESC SHIFT DEL}{ESC SHIFT DEL}";:RETU
RN                                                       <MZ>
```

Listing 1. Einfache Beispiele mit dem SOUND-Befehl. (Schluß)

```
0 REM SOUND.3                                   (
c) 1985 by Peter Gerstner 10/10/85                       <RB>
1 REM                                                    <UQ>
2 REM                                                    <UR>
3 REM                                                    <US>
10 DIM LINE$(40),N(50):V0=0:V1=1:V2=2:V3
=3:GOTO 100                                              <TJ>
30 SOUND V0,N(P),10,14:GOTO 50                           <YO>
40 P0=N(P):P1=N(CHORD):P2=N(CHORD-4):P3=
N(CHORD-7)                                               <PV>
42 SOUND V0,P0,10,14:SOUND V1,P1,10,6:SO
UND V2,P2,10,6:SOUND V3,P3,10,6                          <FX>
50 POKE 540,WAIT                                         <VY>
52 IF PEEK(540)<>0 THEN 52                               <OU>
54 SOUND V0,0,0,0:RETURN                                 <GC>
```

Listing 2. Ein Lied zum Mitsingen. Bitte mit dem Namen »D:SOUND.3« auf Diskette speichern.



```
60 FOR OFF=0 TO 3:SOUND OFF,0,0,0:NEXT O
FF:RETURN                                   <KN>
70 P0=N(P):P1=N(CHORD):P2=N(CHORD-4):P3=
N(CHORD-7)                                  <PY>
72 FOR DECAY=8 TO 0 STEP -1:SOUND V0,P0,
10,DECAY:SOUND V1,P1,10,DECAY:SOUND V2,P
2,10,DECAY:SOUND V3,P3,10,DECAY             <XP>
74 NEXT DECAY:RETURN                        <XI>
100 DATA▲0,14,15,16,17,18,19,21,22,23,24
,26,27,29,31,33,35,37,40,42,45,47,50,53,
57,60,64,68,72,76,81,85,91,96               <QY>
110 DATA▲102,108,114,121,128,136,144,153
,162,173,182,193,204,217,230,243,255        <FP>
120 GOSUB 21000:FOR X=0 TO 50:READ IT:N(
X)=IT:NEXT X                                <AB>
200 POKE 82,4:? "▲":?                       <CH>
210 READ LINE$,CHORD,P,WAIT:? LINE$:GOSU
B 40                                        <SB>
220 FOR ME=1 TO 2:READ P,WAIT:GOSUB 30:N
EXT ME:GOSUB 60:WAIT=10:GOSUB 50            <LE>
223 READ CHORD,P,WAIT:GOSUB 40              <ZF>
225 FOR ME=1 TO 2:READ P,WAIT:GOSUB 30:N
EXT ME:GOSUB 60:WAIT=10:GOSUB 50            <LT>
230 READ LINE$,CHORD,P,WAIT:? :? LINE$:G
OSUB 40                                     <YB>
240 FOR ME=1 TO 6:READ P,WAIT:GOSUB 30:N
EXT ME:GOSUB 60:WAIT=30:GOSUB 50            <SS>
250 READ LINE$,CHORD,P,WAIT:? :? LINE$:G
OSUB 40                                     <YF>
260 FOR ME=1 TO 2:READ P,WAIT:GOSUB 30:N
EXT ME:GOSUB 60:WAIT=10:GOSUB 50            <LM>
263 READ CHORD,P,WAIT:GOSUB 40              <ZN>
265 FOR ME=1 TO 2:READ P,WAIT:GOSUB 30:N
EXT ME:GOSUB 60:WAIT=10:GOSUB 50            <MB>
270 READ LINE$,CHORD,P,WAIT:? :? LINE$:G
OSUB 40                                     <YJ>
280 FOR ME=1 TO 4:READ P,WAIT:GOSUB 30:N
EXT ME:GOSUB 60:WAIT=30:GOSUB 50            <RS>
290 READ LINE$,CHORD,P,WAIT:? :? LINE$:G
OSUB 40                                     <YN>
300 FOR ME=1 TO 6:READ P,WAIT:GOSUB 30:N
EXT ME                                      <VF>
320 READ LINE$,CHORD,P,WAIT:? :? LINE$:G
OSUB 40                                     <YA>
330 FOR ME=1 TO 8:READ P,WAIT:GOSUB 30:N
EXT ME                                      <WT>
350 READ LINE$,CHORD,P,WAIT:? :? LINE$:G
OSUB 40                                     <YG>
360 FOR ME=1 TO 2:READ P,WAIT:GOSUB 30:N
EXT ME                                      <TB>
363 READ CHORD,P,WAIT:GOSUB 40              <ZO>
365 FOR ME=1 TO 2:READ P,WAIT:GOSUB 30:N
EXT ME                                      <TQ>
380 READ LINE$,CHORD,P,WAIT:? :? LINE$:G
OSUB 40                                     <YM>
390 READ P,WAIT:GOSUB 30                    <MM>
400 READ CHORD,P,WAIT:GOSUB 40              <YU>
410 READ P,WAIT:GOSUB 30                    <LX>
420 READ CHORD,P,WAIT:GOSUB 40              <YY>
430 READ P,WAIT:GOSUB 30                    <MB>
440 READ CHORD,P,WAIT:GOSUB 40              <ZC>
450 GOSUB 60:WAIT=10:GOSUB 50:FOR DECAY=
15 TO 0 STEP -0.5:SOUND V0,N(1),10,DECAY
:NEXT DECAY                                 <QW>
460 GRAPHICS 18:? #6:? #6;"▲▲▲▲dur▲akkor
de"                                         <JK>
510 FOR ME=1 TO 8:READ CHORD,P,LINE$:POS
ITION ME*2,10-ME:? #6;LINE$:GOSUB 70:NEX
T ME                                        <NV>
530 FOR ME=8 TO 1 STEP -1:READ CHORD,P,L
INE$:POSITION ME*2,10-ME:? #6;LINE$:GOSU
B 70:NEXT ME                                <GH>
540 WAIT=15:GOSUB 50:POSITION 3,11:? #6;
"DRUECKE"                                   <ZB>
550 FOR DECAY=15 TO 0 STEP -0.5:SOUND V0
,N(6),10,DECAY:NEXT DECAY                   <CT>
560 WAIT=15:GOSUB 50:POSITION 11,11:? #6
;"START"                                    <CU>
570 FOR DECAY=15 TO 0 STEP -0.5:SOUND V0
,N(1),10,DECAY:NEXT DECAY                   <UL>
600 SETCOLOR 0,PEEK(20),10:IF PEEK(53279
)<>6 THEN 600                               <JJ>
605 RESTORE 22000                           <CC>
610 FOR L=1 TO 5                            <OV>
620 GOSUB 21000                             <TG>
630 READ X,Y:LIST X,Y                       <DL>
640 GOSUB 20000                             <SY>
650 NEXT L:GOSUB 21000                      <EF>
655 FOR L=1 TO 3                            <OM>
660 READ X,Y:LIST X,Y                       <DR>
665 NEXT L                                  <IB>
670 GOSUB 20000                             <TE>
700 GRAPHICS 18:SETCOLOR 0,1,10:SETCOLOR
 1,11,12:SETCOLOR 3,4,12                    <YO>
710 ? #6:? #6:? #6;"▲▲▲DRUECKE▲option":?
 #6;"▲▲FUER▲WIEDERHOLEN"                    <HR>
720 WAIT=60:GOSUB 50                        <HV>
730 ? #6:? #6;"▲▲▲DRUECKE▲start":? #6;"▲
▲▲▲FUER▲WEITER"                             <FO>
740 IF PEEK(53279)=3 THEN RUN               <MR>
750 IF PEEK(53279)=6 THEN 900               <IW>
760 GOTO 740                                <PY>
900 RUN "D:SOUND.4":REM CASSETTEN BENUET
ZER MUESSEN HIER GRAPHICS 0:END EINGEBEN
 ANSTELLE RUN "D:SOUND.4"                   <GX>
1000 DATA▲Haens-chen▲klein▲--▲ging▲al-le
in                                          <XL>
1010 DATA▲49,18,30,21,30,21,60,42,20,30,
23,30,23,60                                 <JC>
1020 DATA▲in▲A-ta-ris▲SOUND▲hin-ein         <NI>
1030 DATA▲49,25,30,23,30,21,30,20,30,18,
30,18,30,18,60                              <HD>
1040 DATA▲Sound▲mit▲POKE▲--▲das▲klingt▲t
oll                                         <TP>
1050 DATA▲49,30,30,33,30,33,60,42,32,30,
35,30,35,60                                 <ZX>
1060 DATA▲al-les▲wohl▲und▲voll              <GW>
1070 DATA▲49,37,30,33,30,30,30,30,30,37,
120                                         <KC>
1080 DATA▲Reicht▲der▲acht▲Bit▲SOUND▲nich
t▲aus                                       <TB>
1090 DATA▲42,35,30,35,30,35,30,35,30,35,
30,33,30,32,60                              <FM>
1100 DATA▲mach-en▲wir▲sech-zehn▲Bit▲da-r
aus                                         <QE>
1110 DATA▲49,33,15,33,15,33,30,33,15,33,
15,33,15,32,15,30,30,0,60                   <FM>
1120 DATA▲Wir▲sind▲fit▲---▲toll▲der▲Tric
k                                           <HO>
1130 DATA▲49,30,30,33,30,33,60,42,32,30,
35,30,35,60                                 <ZT>
1140 DATA▲Vielen▲Dank▲dem▲PO-KEY▲Chip▲..
..                                          <BC>
1150 DATA▲49,25,15,26,15                    <ON>
1160 DATA▲44,28,30,32,30                    <GP>
1170 DATA▲42,26,30,30,30                    <DA>
1180 DATA▲49,25,90                          <GX>
1200 DATA▲49,37,C                           <HB>
1210 DATA▲47,35,D                           <FT>
1220 DATA▲45,33,E                           <EL>
1230 DATA▲44,32,F                           <EE>
1240 DATA▲42,30,G                           <CW>
1250 DATA▲40,28,A                           <CF>
1260 DATA▲38,26,B                           <FC>
1270 DATA▲37,25,C                           <EV>
1300 DATA▲37,25,c                           <ZA>
1310 DATA▲38,26,b                           <ZN>
1320 DATA▲40,28,a                           <WW>
1330 DATA▲42,30,g                           <XT>
1340 DATA▲44,32,f                           <ZH>
1350 DATA▲45,33,e                           <ZU>
1360 DATA▲47,35,d                           <BI>
1370 DATA▲49,37,c                           <CW>
20000 POSITION 2,20:POKE 201,4:POKE 752,
1                                           <PR>
20010 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL E}"                              <AT>
20020 ? ,"{SHIFT =}▲Druecke▲START▲fuer▲w
eiter▲{SHIFT =}"                            <PW>
20030 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
```

Listing 2. Ein Lied zum Mitsingen. (Fortsetzung)

```
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL C}"                             <BW>
20040 IF PEEK(53279)<>6 THEN 20040         <VY>
20050 RETURN                               <TS>
21000 GRAPHICS 0:SETCOLOR 2,9,0:SETCOLOR
 4,9,0:SETCOLOR 1,9,12:POKE 752,1:POKE 8
2,2:POKE 83,39:POKE 201,3                  <OI>
21010 POSITION 2,0:? ,"{CTL Q}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL ·R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL E}
"                                          <JA>
21020 ? ,"{SHIFT =}▲BEGLEITUNG▲MIT▲DUR▲A
KKORDEN▲{SHIFT =}"                         <KJ>
21030 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL·C}"               <HM>
21040 POKE 77,0:RETURN                     <AK>
22000 DATA▲0,10,100,120,21000,21040,100,
210,30,74,210,220,40,54,1000,1010          <RZ>
```

Listing 2. Ein Lied zum Mitsingen. (Schluß)

```
0 REM SOUND.4                              (
c) 1985 by Peter Gerstner 11/10/85         <UE>
1 REM                                      <UQ>
2 REM                                      <UR>
3 REM                                      <US>
10 DIM TYPE$(40),N(50):V0=0:V1=1:V2=2:V3
=3:POKE 82,2:? "{ESC CTL -}":GOTO 100      <PD>
50 POKE 540,WAIT                           <VY>
52 IF PEEK(540)<>0 THEN 52                 <OU>
54 RETURN                                  <ML>
60 FOR OFF=0 TO 3:SOUND OFF,0,0,0:NEXT O
FF:RETURN                                  <KN>
70 LINE=LEN(TYPE$):POSITION HOR,VER        <NG>
72 FOR ME=1 TO LINE:? TYPE$(ME,ME);:IF T
YPE$(ME,ME)="▲" THEN 76                    <DL>
74 REM SOUND 0,25,4,6:FOR DECAY=6 TO 0 S
TEP -0.5:SOUND 0,10,0,DECAY:NEXT DECAY     <AG>
76 NEXT ME:RETURN                          <NZ>
100 DATA▲14,15,16,17,18,19,21,22,23,24,2
6,27,29,31,33,35,37,40,42,45,47,50,53,57
,60,64,68,72,76,81,85,91,96                <WW>
110 DATA▲102,108,114,121,128,136,144,153
,162,173,182,193,204,217,230,243,255       <FP>
120 GOSUB 21000:FOR X=1 TO 50:READ IT:N(
X)=IT:NEXT X                               <AY>
130 TYPE$="Bitte▲den▲Begleittext▲fuer▲SO
UND.4▲":HOR=3:VER=9:GOSUB 70               <CZ>
140 TYPE$="lesen▲und▲erst▲die▲START-Tast
e▲druecken":HOR=1:VER=11:GOSUB 70          <QX>
150 TYPE$="wenn▲es▲im▲Text▲verlangt▲wird
.":HOR=5:VER=13:GOSUB 70                   <UA>
160 GOSUB 20000:GOSUB 21000                <XP>
190 TYPE$="DUR▲▲AKKORDE:":HOR=14:VER=6:G
OSUB 70:GOSUB 50:? :? :LIST 200,210        <HF>
200 FOR X=13 TO 50:SOUND 0,N(X),10,14:SO
UND 1,N(X-4),10,6:SOUND 2,N(X-7),10,6:SO
UND 3,N(X-12),10,6                         <XG>
210 GOSUB 60:NEXT X                        <PB>
215 IF LINE<>200 THEN ? :LIST 220,230      <FU>
220 FOR X=50 TO 13 STEP -1:SOUND 0,N(X),
10,14:SOUND 1,N(X-4),10,6:SOUND 2,N(X-7)
,10,6:SOUND 3,N(X-12),10,6                 <KH>
230 GOSUB 60:NEXT X                        <PF>
240 LINE=200:GOSUB 22000                   <RR>
250 TYPE$="MOLL▲AKKORDE:":HOR=14:VER=6:G
OSUB 70:GOSUB 50:? :LIST 260,270           <IK>
260 FOR X=13 TO 50:SOUND 0,N(X),10,14:SO
UND 1,N(X-3),10,6:SOUND 2,N(X-7),10,6:SO
UND 3,N(X-12),10,6                         <VT>
270 GOSUB 60:NEXT X                        <PN>
280 IF LINE<>260 THEN ? :LIST 290,300      <PA>
290 FOR X=50 TO 13 STEP -1:SOUND 0,N(X),
10,14:SOUND 1,N(X-3),10,6:SOUND 2,N(X-7)
,10,6:SOUND 3,N(X-12),10,6                 <IO>
300 GOSUB 60:NEXT X                        <PA>
310 LINE=260:GOSUB 22000                   <UA>
320 X=49                                   <AV>
330 TYPE$="C▲DUR":HOR=17:VER=6:GOSUB 70:
GOSUB 50:? :LIST 340:?                     <AB>
340 X=49:SOUND 0,N(X),10,14:SOUND 1,N(X-
4),10,6:SOUND 2,N(X-7),10,6:SOUND 3,N(X-
12),10,6                                   <AC>
350 WAIT=60:GOSUB 50:GOSUB 60              <HZ>
360 IF LINE<>340 THEN TYPE$="C▲MOLL":HOR
=17:VER=12:GOSUB 70:GOSUB 50:? :LIST 370
:?                                         <DU>
370 X=49:SOUND 0,N(X),10,14:SOUND 1,N(X-
3),10,6:SOUND 2,N(X-7),10,6:SOUND 3,N(X-
12),10,6                                   <YT>
380 WAIT=60:GOSUB 50:GOSUB 60              <IF>
390 LINE=340:GOSUB 22000                   <UE>
400 TYPE$="OK,▲zurueck▲zum▲normalen▲SOUN
D":HOR=5:VER=6:GOSUB 70                    <XT>
410 TYPE$="Befehl▲und▲machen▲einen▲Vergl
eich.":HOR=3:VER=8:GOSUB 70:GOSUB 50       <UT>
420 TYPE$="Die▲tiefste▲Note▲die▲wir▲mit"
:HOR=6:VER=10:GOSUB 70                     <RM>
430 TYPE$="mit▲DISTORTION▲10▲bekommen▲ko
ennen,":HOR=3:VER=12:GOSUB 70:GOSUB 50     <WW>
435 TYPE$="ist▲ein▲tiefes▲'B'.":HOR=11:V
ER=14:GOSUB 70:GOSUB 50                    <LT>
440 TYPE$="SOUND▲V0,255,10,8":HOR=12:VER
=16:GOSUB 70:GOSUB 60:SOUND V0,255,10,8:
GOSUB 50:GOSUB 60                          <FV>
450 GOSUB 20000:GOSUB 21000                <XQ>
500 TYPE$="Vergleichen▲wir▲nun▲zwei▲Soun
ds...":HOR=4:VER=6:GOSUB 70:GOSUB 50       <XF>
510 TYPE$="SOUND▲V0,255,10,8":HOR=12:VER
=8:GOSUB 70:GOSUB 50                       <JE>
520 SOUND V0,255,10,8:GOSUB 50:GOSUB 60:
GOSUB 50                                   <YS>
530 TYPE$="SOUND▲V0,33,12,8":HOR=12:VER=
10:GOSUB 70:GOSUB 50                       <MF>
540 SOUND V0,33,12,8:GOSUB 50:GOSUB 60:G
OSUB 50                                    <VJ>
550 TYPE$="Die▲Tonqualitaet▲ist▲verschie
den▲aber":HOR=2:VER=12:GOSUB 70            <AD>
560 TYPE$="die▲aktuelle▲PITCH▲ist▲das▲gl
eiche▲B.":HOR=2:VER=14:GOSUB 70:GOSUB 50   <UZ>
570 POSITION 12,8:? "SOUND▲V0,255,10,8":
POSITION 12,10:? "SOUND▲V0,33,12,8":SOUN
D V0,255,10,8                              <KF>
580 GOSUB 50:GOSUB 60:GOSUB 50:POSITION
12,8:? "SOUND▲V0,255,10,8"                 <YO>
590 POSITION 12,10:? "SOUND▲V0,33,12,8":
SOUND V0,33,12,8:GOSUB 50:GOSUB 60:GOSUB
 50                                        <IO>
600 POSITION 12,8:? "SOUND▲V0,255,10,8":
SOUND V0,255,10,8:POSITION 12,10:? "SOUN
D▲V1,33,12,8":SOUND V1,33,12,8             <NS>
610 GOSUB 50:GOSUB 60:LINE=570:GOSUB 220
00                                         <VZ>
620 TYPE$="Wir▲koennen▲mit▲DISTORTION▲12
":HOR=5:VER=6:GOSUB 70                     <RJ>
630 TYPE$="anstatt▲DISTORTION▲10▲noch▲ti
efere":HOR=3:VER=8:GOSUB 70:GOSUB 50       <NE>
635 TYPE$="BASS▲NOTEN▲bekommen.":HOR=11:
VER=10:GOSUB 70:GOSUB 50                   <SP>
640 TYPE$="ZUM▲BEISPIEL:":HOR=14:VER=12:
GOSUB 70:GOSUB 50                          <YQ>
700 DATA▲25,27,28,30,31,33,36,37,40,42,4
5,48,51,52,57,60,63,67,72,75,82,85,90,97
,102                                       <FN>
710 FOR BASS=1 TO 25:READ P0:FOR DECAY=1
5 TO 0 STEP -1                             <AT>
720 POSITION 12,14:? "SOUND▲V0,";P0;",12
,";DECAY;"▲":SOUND V0,P0,12,DECAY:NEXT D
ECAY:NEXT BASS                             <JH>
```

Listing 3. Verschiedene Akkorde. Bitte mit dem Namen »D:SOUND.4« auf Diskette speichern.

```
750 RESTORE 700:LINE=710:GOSUB 22000           <CB>
800 TYPE$="Das naechste Programm erlaubt
 uns":HOR=4:VER=6:GOSUB 70                     <XP>
810 TYPE$="verschiedene Sounds anzuhoere
n, wenn":HOR=3:VER=8:GOSUB 70                  <DK>
820 TYPE$="wir mit den Joystick PITCH, V
OLUME,":HOR=3:VER=10:GOSUB 70                  <XO>
830 TYPE$="und DISTORTION veraendern.":H
OR=7:VER=12:GOSUB 70:GOSUB 20000               <DO>
840 RUN "D:SOUND.5":REM CASSETTEN BENUET
ZER MUESSEN HIER GRAPHICS 0:END EINGEBEN
 ANSTELLE VON RUN "D:SOUND.5"                  <RG>
20000 POSITION 2,20:POKE 201,4:POKE 752,
1                                              <PR>
20010 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL E}"                                 <AT>
20020 ? ,"{SHIFT =} Druecke START fuer w
eiter {SHIFT =}"                               <PW>
20030 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL C}"                                 <BW>
20040 IF PEEK(53279)<>6 THEN POKE 755,3:
POKE 755,2:GOTO 20040                          <VV>
20050 POKE 755,2:RETURN                        <CS>
21000 GRAPHICS 0:SETCOLOR 2,9,0:SETCOLOR
 4,9,0:SETCOLOR 1,9,12:POKE 752,1:POKE 8
2,2:POKE 83,39:POKE 201,8                      <IN>
21010 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL E}"         <JM>
21020 ? ,"{SHIFT =} Der SOUND Befehl {SH
IFT =}"                                        <IV>
21030 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL C}"         <LH>
21040 POKE 77,0:RETURN                         <AK>
22000 POSITION 2,19:POKE 201,5:POKE 752,
1                                              <WW>
22010 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL E}"       <XZ>
22015 ? ,"{SHIFT =} OPTION = Wiederholu
ng {SHIFT =}"                                  <UV>
22020 ? ,"{SHIFT =} START = Weiter
 {SHIFT =}"                                    <JE>
22030 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL C}"       <ZG>
22040 IF PEEK(53279)=6 THEN POKE 755,2:G
OSUB 21000:RETURN                              <TV>
22050 IF PEEK(53279)=3 THEN POP :POKE 75
5,2:POSITION 2,19:? "{ESC SHIFT DEL}{ESC
 SHIFT DEL}{ESC SHIFT DEL}{ESC SHIFT DEL
}";:GOTO LINE                                  <OW>
22060 POKE 755,3:POKE 755,2:GOTO 22040         <XL>
```

Listing 3. Verschiedene Akkorde. (Schluß)

```
0 REM SOUND.5
(c) 1985 by Peter Gerstner 13/10/85            <QJ>
1 REM                                          <UQ>
2 REM                                          <UR>
140 GRAPHICS 2:POKE 752,1:POKE 710,145:P
OKE 712,145:POKE 201,10:? "{ESC CTL -}"        <NX>
150 ? "JOYSTICK auf/ab        = TONHOEHE"      <IZ>
160 ? "JOYSTICK trigger       = VERZERRUN
G"                                             <ZR>
170 ? "JOYSTICK links/rechts = LAUTSTAER
KE"                                            <WV>
180 ? "DRUECKE EINE BELIEBIGE TASTE FUER
 END";                                         <VW>
190 POKE 764,255                               <WZ>
200 VOLUME=2                                   <RO>
210 DISTORTION=10                              <KQ>
220 DELAY=320                                  <AQ>
230 IF STICK(0)=14 THEN PITCH=PITCH+1:IF
 PITCH>255 THEN PITCH=255                      <XK>
235 IF STICK(0)=13 THEN PITCH=PITCH-1:IF
 PITCH<0 THEN PITCH=0                          <DP>
240 IF STRIG(0)=0 THEN DISTORTION=DISTOR
TION+2:GOSUB DELAY                             <YM>
250 IF DISTORTION=16 THEN DISTORTION=0         <PM>
260 IF STICK(0)=7 THEN VOLUME=VOLUME+1:I
F VOLUME>15 THEN VOLUME=15                     <BX>
270 IF STICK(0)=11 THEN VOLUME=VOLUME-1:
IF VOLUME<0 THEN VOLUME=0                      <FU>
280 IF PEEK(764)<>255 THEN GRAPHICS 0:PO
KE 764,255:POKE 752,0:RUN "D:SOUND.6"          <FA>
290 POSITION 2,4:? #6;"sound 0,";PITCH;"
,";DISTORTION;",";VOLUME;"    "                <CM>
300 SOUND 0,PITCH,DISTORTION,VOLUME            <ZB>
310 SETCOLOR 0,DISTORTION,10:SETCOLOR 1,
VOLUME,10:POKE 77,0:GOTO 230                   <SN>
320 FOR WAIT=1 TO 100:NEXT WAIT:RETURN         <YJ>
330 REM CASSETTEN BENUETZER MUESSEN IN Z
EILE 280 'RUN "D:SOUND.6" ' AENDERN IN '
 GRAPHICS 0:END '                              <QA>
```

Listing 4. Experimentieren mit dem SOUND-Befehl. Bitte mit dem Namen »D:SOUND.5« auf Diskette speichern.

```
0 REM SOUND.6                              (
c) 1985 by Peter Gerstner 03/06/85             <LN>
1 REM                                          <UQ>
2 REM                                          <UR>
3 REM                                          <US>
10 V0=0:V1=1:V2=2:V3=3:POKE 82,2:POKE 83
,39:GOTO 2000                                  <TX>
50 POKE 540,WAIT                               <VY>
52 IF PEEK(540)<>0 THEN 52                     <OU>
54 FOR OFF=0 TO 3:SOUND OFF,0,0,0:NEXT O
FF:RETURN                                      <KU>
89 REM                                         <KN>
90 REM *** MACHINE GUN ***                     <QM>
91 REM                                         <JY>
100 FOR SHOT=1 TO 12:FOR VOL=15 TO 0 STE
P -5:SOUND V0,80,0,VOL:SOUND V1,60,0,VOL       <VA>
110 SOUND V2,200,4,VOL:SOUND V3,10,4,VOL
:NEXT VOL:GOSUB 54:NEXT SHOT                   <GH>
120 RETURN                                     <MP>
189 REM                                        <BZ>
190 REM *** SURF/WAVES ***                     <TY>
191 REM                                        <BD>
200 FOR P0=10 TO 2 STEP -0.02:VOL=P0/2:S
OUND V0,P0,8,VOL:SOUND V1,P0+1,8,VOL           <DT>
210 SOUND V2,P0+2,8,VOL:SOUND V3,RND(0)*
3,8,VOL                                        <XY>
220 FOR P0=3 TO 12 STEP 0.02:VOL=P0/2:SO
UND V0,P0,8,VOL:SOUND V1,P0+1,8,VOL            <TC>
230 SOUND V2,P0+2,8,VOL:SOUND V3,RND(0)*
3,8,VOL:NEXT P0                                <SK>
240 FOR P0=10 TO 2 STEP -0.02:VOL=P0/2:S
OUND V0,P0,8,VOL:SOUND V1,P0+1,8,VOL           <EB>
250 SOUND V2,P0+2,8,VOL:SOUND V3,RND(0)*
3,8,VOL:NEXT P0:GOSUB 54:RETURN                <MB>
289 REM                                        <CA>
290 REM *** LAZERS/PHOTONS ***                 <CQ>
291 REM                                        <BE>
300 FOR SHOT=1 TO 6:FOR P0=0 TO 200 STEP
 10                                            <EN>
```

Listing 5. Verschiedene Sound-Effekte. Bitte mit dem Namen »D:SOUND.6« auf Diskette speichern.

```
310 SOUND V0,P0,0,8:SOUND V1,P0,10,8:SOU
ND V2,P0,12,8:SOUND V3,P0,4,8                <AJ>
320 NEXT P0:NEXT SHOT:GOSUB 54:RETURN        <YW>
389 REM                                      <CB>
390 REM *** POLICE/FIRE SIREN ***            <DP>
391 REM                                      <BF>
400 FOR P0=200 TO 50 STEP -1:SOUND V0,P0
,10,8:SOUND V1,P0+2,10,6:SOUND V2,P0+4,1
0,2:SOUND V3,P0+6,10,2:NEXT P0               <TG>
420 FOR P0=50 TO 160 STEP 0.2:SOUND V0,P
0,10,8:SOUND V1,P0+2,10,6:SOUND V2,P0+4,
10,4:SOUND V3,P0+6,10,2:NEXT P0              <MT>
430 GOSUB 54:RETURN                          <QZ>
489 REM                                      <CC>
490 REM *** AIR RAID SIREN ***               <ZX>
491 REM                                      <BG>
500 FOR LOOP=1 TO 6:FOR P0=1 TO 20:SOUND
 V0,80+P0,12,8:NEXT P0                       <KO>
510 SOUND V0,80,10,12:SOUND V1,100,10,12
:SOUND V2,13,4,12                            <AH>
520 WAIT=30:GOSUB 50:NEXT LOOP               <QX>
530 FOR V=12 TO 0 STEP -0.1:SOUND V0,(20
-V)*10,10,V:SOUND V1,(20-V)*10+20,10,V:S
OUND V2,13,4,V:NEXT V                        <OJ>
540 GOSUB 54:RETURN                          <RC>
589 REM                                      <CD>
590 REM *** TELEPHONE RINGING ***            <BM>
591 REM                                      <BH>
600 FOR RING=1 TO 2:FOR LOUD=1 TO 35:SOU
ND V0,20,10,8:SOUND V1,1,2,8                 <MN>
610 FOR LOOP=1 TO 2:SOUND V0,25,10,8:SOU
ND V1,0,2,8:NEXT LOOP:SOUND V0,0,0,0:SOU
ND V1,0,0,0:NEXT LOUD                        <IL>
620 FOR V=7 TO 0 STEP -0.2:SOUND V0,20,1
0,V:SOUND V1,0,2,V:NEXT V                    <CG>
630 WAIT=90:GOSUB 50:NEXT RING:GOSUB 54:
RETURN                                       <OH>
689 REM                                      <CE>
690 REM *** WHISTLING BOMB ***               <CI>
691 REM                                      <BI>
700 FOR P0=0 TO 150:SOUND 0,P0,10,P0/15+
2:NEXT P0                                    <SA>
710 FOR P0=0 TO 240 STEP 5:VOL=14-P0/20:
SOUND V0,P0,0,VOL:SOUND V1,P0,8,VOL          <FC>
720 SOUND V2,P0+15,2,VOL:NEXT P0:GOSUB 5
4:RETURN                                     <CC>
789 REM                                      <CF>
790 REM *** SPACE SHIP ***                   <LF>
791 REM                                      <BJ>
800 SOUND V2,0,8,2:FOR VOL=1 TO 15 STEP
0.1:SOUND V0,25,4,VOL:SOUND V1,13,4,VOL:
NEXT VOL                                     <SH>
810 FOR VOL=14 TO 0 STEP -0.1:SOUND V0,2
5,4,VOL:SOUND V1,13,4,VOL:NEXT VOL           <GK>
820 GOSUB 54:RETURN                          <RB>
889 REM                                      <CG>
890 REM *** SPACE ECHO ***                   <VN>
891 REM                                      <BK>
900 FOR VOL=15 TO 0 STEP -0.2:FOR P0=0 T
O 5:SOUND V0,P0,2,VOL:SOUND V1,P0+1,2,VO
L:NEXT P0                                    <UF>
910 FOR P1=VOL*10 TO VOL STEP -10:SOUND
V0,P1,10,VOL:SOUND V1,P1+VOL,10,VOL:NEXT
 P1:NEXT VOL                                 <DG>
920 RETURN                                   <MX>
989 REM                                      <CH>
990 REM *** DOOR BELL ***                    <TK>
991 REM                                      <BL>
1000 FOR VOL=15 TO 0 STEP -0.5:SOUND V0,
29,10,VOL:SOUND V1,30,10,VOL:NEXT VOL        <FS>
1010 FOR VOL=15 TO 0 STEP -0.5:SOUND V0,
35,10,VOL:SOUND V1,36,10,VOL:NEXT VOL        <PA>
1020 RETURN                                  <OZ>
1089 REM                                     <VG>
1090 REM *** BUZZER ***                      <HA>
1091 REM                                     <UD>
1100 SOUND 0,40,6,10                         <HL>
1110 FOR DELAY=1 TO 400                      <XH>
1120 NEXT DELAY                              <CZ>
1130 SOUND 0,0,0,0:RETURN                    <DL>
1189 REM                                     <VI>
1190 REM *** GALAXY CANTINA MUSIC ***        <YW>
1191 REM                                     <UF>
1200 FOR I=0 TO 3                            <WL>
1210 FOR X=15 TO 0 STEP -0.2                 <YO>
1220 SOUND 0,X,12,8                          <TM>
1230 FOR DELAY=1 TO 2:NEXT DELAY             <TL>
1240 NEXT X                                  <LO>
1250 NEXT I:SOUND 0,0,0,0:RETURN             <BO>
1289 REM                                     <VK>
1290 REM *** THUNDER ***                     <FK>
1291 REM                                     <UH>
1300 FOR P0=1 TO 3                           <IX>
1310 Y=INT(255*RND(1)+20)                    <ZY>
1320 X=RND(1)*75                             <BG>
1330 FOR P1=1 TO Y                           <JZ>
1340 SOUND 0,P1,8,15                         <CW>
1350 NEXT P1                                 <EZ>
1360 FOR DELAY=1 TO X:NEXT DELAY             <YO>
1370 NEXT P0:SOUND 0,0,0,0:RETURN            <BY>
1389 REM                                     <VM>
1390 REM *** OLD AIRPLANE ***                <IO>
1391 REM                                     <UJ>
1400 FOR X=1 TO 400                          <TU>
1410 SOUND 0,99,10,8                         <RR>
1420 SOUND 0,0,0,0                           <HK>
1430 NEXT X:RETURN                           <KH>
1489 REM                                     <VO>
1490 REM *** PLANE CRASHING ***              <NC>
1491 REM                                     <UL>
1500 FOR X=255 TO 0 STEP -1                  <HW>
1510 SOUND 0,X,8,8                           <NJ>
1520 FOR I=1 TO 5                            <YR>
1530 NEXT I:NEXT X                           <GO>
1540 FOR X=15 TO 0 STEP -1                   <CN>
1550 SOUND 0,25,16,X                         <OX>
1560 FOR I=1 TO 20                           <EK>
1570 NEXT I:NEXT X                           <HA>
1580 RETURN                                  <QB>
1999 STOP                                    <DW>
2000 REM *** MENU OPTIONS ***                <XO>
2010 GRAPHICS 0:SETCOLOR 2,15,0:POKE 752
,1:POKE 201,10:?                             <EN>
2020 ? ,"{CTL Q}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL E}"                                    <VE>
2030 ? ,"{SHIFT =}▲SOUND▲EFFEKTE▲{SHIFT
=}"                                          <ED>
2040 ? ,"{CTL Z}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL C}":POKE 201,9:?                       <SV>
2050 POKE 201,5                              <MJ>
2100 ? ,"▲<1>▲MACHINE▲GUN"                   <AH>
2110 ? ,"▲<2>▲SURF▲WAVES"                    <QO>
2120 ? ,"▲<3>▲LAZER▲FIRE"                    <RE>
2130 ? ,"▲<4>▲POLICE▲SIREN"                  <MO>
2140 ? ,"▲<5>▲AIR▲RAID▲SIREN"                <FK>
2150 ? ,"▲<6>▲TELEPHONE▲RINGING"             <ON>
2160 ? ,"▲<7>▲WHISTLING▲BOMB"                <NN>
2170 ? ,"▲<8>▲SPACE▲SHIP"                    <RY>
2180 ? ,"▲<9>▲SPACE▲ECHO"                    <BE>
2190 ? ,"<10>▲DOOR▲BELL"                     <BO>
2200 ? ,"<11>▲BUZZER"                        <UM>
2210 ? ,"<12>▲GALAXY▲CANTINA▲MUSIC"          <HW>
2220 ? ,"<13>▲THUNDER"                       <RG>
2230 ? ,"<14>▲OLD▲AIRPLANE"                  <IU>
2240 ? ,"<15>▲PLANE▲CRASHING"                <JZ>
2500 ? ,"<16>▲RUN▲LAST▲PROGRAM"              <IW>
3000 POKE 53279,0:? :? ,"▲▲▲▲▲WELCHE▲NR.
";:TRAP 9000:INPUT CHOICE:TRAP 40000         <XW>
3010 CHOICE=INT(CHOICE):IF CHOICE<1 OR C
HOICE>16 THEN 9000                           <ZM>
3020 IF CHOICE=16 THEN RUN "D:SOUND.7"       <SC>
3030 GRAPHICS 0:SETCOLOR 2,CHOICE,0:? :?
 :? :? :LIST CHOICE*100-10,CHOICE*100+80     <BB>
3040 GOSUB CHOICE*100:? :? :? "▲▲▲DURECK
E▲EINE▲TASTE▲FUER▲MENUE";:POKE 764,255       <CS>
3050 IF PEEK(764)=255 THEN 3050              <XC>
3060 POKE 764,255:GOTO 2000                  <SK>
9000 RUN                                     <XW>
9010 REM CASSETTEN BENUETZER MUESSEN IN
ZEILE 3020  'RUN "D:SOUND.7"' AENDERN IN
 'GRAPHICS 0:END'                            <CG>
```

Listing 5. Verschiedene Sound-Effekte. (Schluß)

```
0 REM SOUND.7                              (
c) 1985 by Peter Gerstner 26/11/85          <OB>
1 REM                                       <UQ>
2 REM                                       <UR>
3 REM                                       <US>
100 GOTO 10000                              <JC>
500 POSITION 1,0:FOR ME=0 TO 8:POSITION
6,ME+2:? #6;53760+ME;"=";N(ME);"▲▲":NEXT
 ME                                         <DZ>
600 ? #6;"waehle▲speicherzelle";            <SF>
700 POSITION 5,2:? #6;"<ESC CTL +>":Y=2     <ZS>
720 S=STICK(0):IF S=14 AND Y>2 THEN POSI
TION 5,Y:? #6;"▲":Y=Y-1:POSITION 5,Y:? #
6;CHR$(30):GOSUB 3000                       <XN>
730 IF PEEK(764)<>255 THEN 12000            <CH>
740 IF S=13 AND Y<10 THEN POSITION 5,Y:?
 #6;"▲":Y=Y+1:POSITION 5,Y:? #6;CHR$(30)
:GOSUB 3000                                 <RF>
760 IF STRIG(0)=1 THEN 720                  <AG>
800 GC=Y-2                                  <UB>
900 POSITION 0,11:? #6;"▲trigger▲auslass
en▲▲";                                      <EE>
1000 IF STRIG(0)=0 THEN 1000                <YL>
1010 POSITION 5,Y:? #6;"▲"                  <SE>
1020 P=53760+GC:L$=STR$(P):FOR ME=1 TO 5
:IT=ASC(L$(ME,ME)):IT=IT+128:L$(ME,ME)=C
HR$(IT):NEXT ME                             <TQ>
1030 POSITION 6,Y:? #6;L$                   <BP>
1040 POSITION 0,11:? #6;"▲▲▲▲aender▲poke
▲▲▲"                                        <NN>
2000 S=STICK(0):IF S=14 AND N(GC)<255 TH
EN N(GC)=N(GC)+1                            <PW>
2020 IF PEEK(764)<>255 THEN 12000           <UE>
2100 IF S=13 AND N(GC)>0 THEN N(GC)=N(GC
)-1                                         <QE>
2200 POKE P,N(GC):POSITION 12,Y:? #6;N(G
C);"▲▲"                                     <UU>
2300 IF STRIG(0)=1 THEN 2000                <AJ>
2400 POKE 65,0:GOTO 500                     <QJ>
3000 FOR ME=1 TO 10:POKE 53279,0:POKE 53
279,8:NEXT ME:RETURN                        <LM>
10000 GRAPHICS 18:DIM N(8),L$(5):SOUND 0
,0,0,0:POKE 710,154:POKE 709,204            <JM>
10010 REM * LITERALS IN LINE 10100 ARE I
NVERSE VIDEO                                <VC>
10100 ? #6;"▲▲▲▲SOUND▲EDITOR":? #6;"▲▲▲▲
************":POKE 764,255                  <SZ>
11000 FOR ME=0 TO 8:N(ME)=0:NEXT ME:CLOS
E #1:OPEN #1,4,0,"K:":GOTO 500              <US>
12000 FOR A=0 TO 3:SOUND A,0,0,0:NEXT A:
POKE 65,3                                   <GG>
12010 RUN "D:SOUND.8"                       <WO>
12020 REM                                   <NW>
12030 REM Cassetten Benutzer Zeile 12010
 aendern in 'GRAPHICS 0:END'                <FZ>
```

Listing 6. Sound-Editor. Bitte mit dem Namen »D:SOUND.7« auf Diskette speichern.

```
0 REM SOUND.8                               <WR>
1 REM                                       <UQ>
2 REM                                       <UR>
3 GRAPHICS 1+16                             <IJ>
4 SETCOLOR 0,4,6:SETCOLOR 1,0,12:SETCOLO
R 2,11,4:GOSUB 2000                         <XS>
5 FOR I=0 TO 3:SOUND 0,100,10,0:NEXT I      <XL>
10 DIM B(3)                                 <BH>
20 B(1)=128:B(2)=0:B(3)=128                 <ZD>
40 POKE 53768,4:SO=130:POKE 53763,130       <KP>
50 BC=3:RESTORE 1000                        <TJ>
100 READ T:IF T=-1 THEN RESTORE 1100:REA
D T                                         <UK>
105 IF PEEK(53279)=6 THEN GRAPHICS 0:NEW
                                            <QK>
110 BC=BC+1:IF BC=4 THEN BC=1               <KR>
115 SO=SO-10:IF SO=0 THEN SO=120            <XC>
120 C=1536:POKE 53762,SO:POKE 53767,160:
TA=1536                                     <QD>
127 IF B(BC)<>0 THEN C=53767:POKE 53766,
B(BC)                                       <OI>
130 IF T<>0 THEN POKE 53760,T:POKE 53764
,T+1:TA=53761                               <YW>
140 FOR I=14 TO 2 STEP -2:POKE C,192+I:P
OKE TA,160+I:POKE TA+4,160+I:NEXT I         <IQ>
150 GOTO 100                                <LQ>
1000 DATA▲0,0,0,0,0,0,0,0,0,0,0,0           <BL>
1010 DATA▲0,0,0,0,0,0,0,0,0,0,0,0           <BO>
1100 DATA▲162,81,81,81,81,81,91,81,81,81
,81,81                                      <DR>
1110 DATA▲102,81,81,81,81,81,108,81,81,8
1,81,81                                     <KE>
1120 DATA▲121,81,81,81,81,81,121,91,136,
102,144,108                                 <CC>
1130 DATA▲162,121,144,108,136,102,144,10
8,162,121,182,136                           <VQ>
1140 DATA▲162,121,144,108,136,102,144,10
8,162,121,182,136                           <VT>
1150 DATA▲204,144,217,162,243,204,243,18
2,217,162,204,217                           <ET>
1152 DATA▲162,121,144,108,136,102,144,10
8,162,121,182,136                           <WE>
1154 DATA▲162,121,144,108,136,102,144,10
8,162,121,182,136                           <WM>
1160 DATA▲108,0,108,108,0,121,0,0,0,0,0,
0                                           <WS>
1170 DATA▲136,0,136,136,0,144,0,0,0,0,0,
0                                           <EC>
1180 DATA▲162,0,162,0,162,0,144,0,144,0,
144,0                                       <MZ>
1190 DATA▲136,0,136,0,144,0,162,0,0,0,0,
0                                           <FI>
1200 DATA▲108,0,108,108,0,121,0,0,0,0,0,
0                                           <WC>
1210 DATA▲136,0,136,136,0,144,0,0,0,0,0,
0                                           <DM>
1220 DATA▲162,0,162,0,0,162,144,0,144,0,
0,144                                       <OJ>
1230 DATA▲136,0,136,144,0,162,0,0,0,0,0,
0                                           <CS>
1240 DATA▲-1                                <EU>
2000 POSITION 0,3:? #6;"▲VIELEN▲DANK▲FUE
R'S"                                        <AA>
2010 ? #6:? #6;"▲MITMACHEN▲UND▲VIEL"        <CR>
2020 ? #6:? #6;"▲▲ERFOLG▲IM▲MUSIK"          <IE>
2030 ? #6:? #6;"▲▲▲PROGRAMMIEREN"           <HP>
2040 ? #6:? #6;"▲▲▲▲▲▲WUENSCHT"             <WY>
2050 ? #6:? #6;"▲▲▲▲▲▲▲▲DIR"                <DE>
2060 ? #6:? #6:? #6;"▲▲▲peter▲gerstner"     <CY>
2070 POSITION 0,21:? #6;"▲▲▲▲START▲=▲END
E"                                          <AH>
2080 RETURN                                 <PS>
```

Listing 7. »Oxygene« von Jean-Michel Jarre. Bitte mit dem Namen »D:SOUND.8« auf Diskette speichern.

```
0 REM (c) 1985 by Kemal Ezcan               <RY>
1 REM                                       <UQ>
2 REM                                       <UR>
3 REM                                       <US>
10 GOSUB 30000                              <MT>
15 POKE 53768,1                             <PA>
30 SOUND 0,0,10,10:SOUND 1,0,10,10:SOUND
 2,0,10,10:SOUND 3,0,10,10                  <OV>
50 C=C+1:IF C>8 THEN C=1:READ BD,RD,SD      <BO>
60 ST=ASC(S$(C+SD,C+SD))                    <XL>
70 BT=ASC(B$(C+BD,C+BD)):SOUND 3,BT,12,8    <JI>
75 IF PEEK(53279)=6 THEN GRAPHICS 0:END     <XH>
80 RDIS=1:A=ASC(R$(C+RD,C+RD)):IF A<>0 T
HEN RT=A:RL=12                              <IH>
90 READ T,E:IF T=-1 THEN 200                <QN>
95 SOUND 0,T,10,12                          <SF>
100 FOR I=15 TO 4 STEP -1.6:SOUND 1,ST,1
2,I:SOUND 2,RT,8,RL:RL=RL-0.9*(RL>=0.6):
NEXT I                                      <SA>
101 SOUND 2,0,3,10                          <GA>
105 IF E=0 THEN SOUND 0,0,0,0               <HM>
110 GOTO 50                                 <QS>
200 FOR I=15 TO 4 STEP -1:SOUND 0,91,10,
I:SOUND 1,121,10,I:SOUND 2,72,10,I:SOUND
 3,0,0,0:NEXT I                             <VT>
202 FOR I=15 TO 4 STEP -1:SOUND 0,91,10,
I:SOUND 1,121,10,I:SOUND 2,72,10,I:SOUND
 3,0,0,0:NEXT I                             <VZ>
204 FOR I=15 TO 4 STEP -1:SOUND 0,96,10,
```

Listing 8. »Digi Loo-Digi Ley« – Top-Musik auf dem Atari

```
I:SOUND 1,121,10,I:SOUND 2,81,10,I:SOUND
 3,0,0,0:NEXT I                           <DC>
206 FOR I=15 TO 0 STEP -0.8:SOUND 0,91,1
0,I:SOUND 1,121,10,I:SOUND 2,72,10,I:SOU
ND 3,0,0,0:NEXT I                         <AM>
210 GRAPHICS 0:END                        <HT>
220 STOP                                  <EO>
1000 DATA▲56,0,56,0,1,0,1,0,1,0,1,0,1,0,
1,0,1,0,1                                 <FQ>
1010 DATA▲56,0,56,72,0,72,0,81,0,72,0,0,
1,0,1,0,1,0,1                             <WC>
1020 DATA▲56,0,56,0,1,0,1,0,1,0,1,0,1,0,
1,0,1,0,1                                 <FW>
1030 DATA▲0,0,56,72,0,72,0,81,0,72,0,0,1
,0,1,0,1,0,1                              <YK>
1040 DATA▲0,8,56,0,1,0,1,0,1,0,1,0,1,0,1
,0,1,0,1                                  <UU>
1045 REM VORSPEL                          <PR>
1050 DATA▲56,0,0,72,0,72,0,72,0,81,1,81,
0,72,1,72,0,60,0                          <HU>
1060 DATA▲56,0,0,33,0,33,0,35,0,45,1,45,
0,40,1,40,0,0,1                           <BP>
1070 DATA▲56,0,0,60,0,60,0,60,0,60,1,60,
0,91,1,91,0,45,0                          <BH>
1080 DATA▲56,0,8,22,0,22,0,22,0,22,0,23,
1,23,0,26,1,26,0                          <KD>
1090 DATA▲56,0,16,60,0,60,0,60,0,72,1,72
,0,81,0,91,0,81,1                         <OH>
1100 DATA▲56,0,24,81,1,81,1,81,1,81,0,68
,1,68,1,68,1,68,0                         <IW>
1110 DATA▲56,0,16,60,0,60,0,60,0,72,1,72
,0,81,0,91,0,60,1                         <HE>
1120 DATA▲56,0,32,60,1,60,1,60,1,60,1,60
,0,45,0,47,0,60,1                         <CS>
1130 DATA▲56,8,40,60,1,60,1,60,1,60,1,60
,1,60,0,72,0,68,0                         <NO>
1140 REM REFRAIN                          <YQ>
1150 DATA▲16,0,16,60,1,60,0,60,0,72,0,53
,1,53,0,53,0,72,0                         <EI>
1160 DATA▲24,0,24,68,0,72,0,68,0,60,1,60
,1,60,0,72,0,68,0                         <XX>
1170 DATA▲0,0,0,60,0,60,0,60,0,60,0,60,0
,91,1,91,0,53,1                           <WF>
1180 DATA▲32,0,32,53,1,53,1,53,1,53,1,53
,1,53,1,53,1,53,0                         <MJ>
1190 DATA▲8,0,8,53,0,53,0,53,0,53,0,47,1
,47,0,47,1,47,0                           <DM>
1200 DATA▲0,0,0,45,0,60,1,60,0,60,1,60,1
,60,1,60,0,72,0                           <NO>
1210 DATA▲48,0,48,68,0,72,0,68,1,68,1,68
,0,53,1,53,0,53,1                         <TQ>
1215 REM {CTL Q}{CTL R}{CTL R}{CTL R}1{C
TL R}{CTL R}{CTL R}:{CTL V}               <XD>
1220 DATA▲40,8,40,53,1,53,0,60,1,60,1,60
,1,60,0,72,0,68,0                         <MV>
1230 REM REFRAIN                          <YP>
1240 DATA▲16,0,16,60,1,60,0,60,0,72,0,53
,1,53,0,53,0,72,0                         <EH>
1250 DATA▲24,0,24,68,0,72,0,68,0,60,1,60
,1,60,0,72,0,68,0                         <XW>
1260 DATA▲0,0,0,60,0,60,0,60,0,60,0,60,0
,91,1,91,0,53,1                           <WE>
1270 DATA▲32,0,32,53,1,53,1,53,1,53,1,53
,1,53,1,53,1,53,0                         <MI>
1280 DATA▲8,0,8,53,0,53,0,53,0,53,0,47,1
,47,0,47,1,47,0                           <DL>
1290 DATA▲0,0,0,45,0,60,1,60,0,60,1,60,1
,60,1,60,0,72,0                           <OP>
1300 DATA▲48,0,48,68,0,72,0,68,1,68,1,68
,0,53,1,53,0,53,1                         <TP>
1310 REM {CTL Q}{CTL R}{CTL R}{CTL R}2{C
TL R}{CTL R}{CTL R}                       <SP>
1320 DATA▲40,0,40,53,1,53,0,60,1,60,1,60
,0,45,0,47,0,60,1                         <BM>
1330 DATA▲40,0,40,60,1,60,1,60,1,60,1,60
,1,60,1,60,1,60,0                         <NJ>
1340 REM WIEDERHOLUNG                     <WA>
1350 DATA▲56,0,0,72,0,72,0,72,0,81,1,81,
0,72,1,72,0,60,0                          <IA>
1360 DATA▲56,0,0,33,0,33,0,35,0,45,1,45,
0,40,1,40,0,0,1                           <BV>
1370 DATA▲56,0,0,60,0,60,0,60,0,60,1,60,
0,91,1,91,0,45,0                          <BN>
```

```
1380 DATA▲56,0,8,22,0,22,0,22,0,22,0,23,
1,23,0,26,1,26,0                          <KJ>
1390 DATA▲56,0,16,60,0,60,0,60,0,72,1,72
,0,81,0,91,0,81,1                         <ON>
1400 DATA▲56,0,24,81,1,81,1,81,1,81,0,68
,1,68,1,68,1,68,0                         <JC>
1410 DATA▲56,0,16,60,0,60,0,60,0,72,1,72
,0,81,0,91,0,60,1                         <HK>
1420 DATA▲56,0,32,60,1,60,1,60,1,60,1,60
,0,45,0,47,0,60,1                         <CY>
1430 DATA▲56,8,40,60,1,60,1,60,1,60,1,60
,1,60,0,72,0,68,0                         <NU>
1440 REM REFRAIN                          <YW>
1450 DATA▲16,0,16,60,1,60,0,60,0,72,0,53
,1,53,0,53,0,72,0                         <EO>
1460 DATA▲24,0,24,68,0,72,0,68,0,60,1,60
,1,60,0,72,0,68,0                         <YD>
1470 DATA▲0,0,0,60,0,60,0,60,0,60,0,60,0
,91,1,91,0,53,1                           <WL>
1480 DATA▲32,0,32,53,1,53,1,53,1,53,1,53
,1,53,1,53,1,53,0                         <MP>
1490 DATA▲8,0,8,53,0,53,0,53,0,53,0,47,1
,47,0,47,1,47,0                           <DS>
1500 DATA▲0,0,0,45,0,60,1,60,0,60,1,60,1
,60,1,60,0,72,0                           <NU>
1510 DATA▲48,0,48,68,0,72,0,68,1,68,1,68
,0,53,1,53,0,53,1                         <TW>
1515 REM {CTL Q}{CTL R}{CTL R}{CTL R}1{C
TL R}{CTL R}{CTL R}:{CTL V}               <XJ>
1520 DATA▲40,8,40,53,1,53,0,60,1,60,1,60
,1,60,0,72,0,68,0                         <NB>
1530 REM REFRAIN                          <YV>
1540 DATA▲16,0,16,60,1,60,0,60,0,72,0,53
,1,53,0,53,0,72,0                         <EN>
1550 DATA▲24,0,24,68,0,72,0,68,0,60,1,60
,1,60,0,72,0,68,0                         <YC>
1560 DATA▲0,0,0,60,0,60,0,60,0,60,0,60,0
,91,1,91,0,53,1                           <WK>
1570 DATA▲32,0,32,53,1,53,1,53,1,53,1,53
,1,53,1,53,1,53,0                         <MO>
1580 DATA▲8,0,8,53,0,53,0,53,0,53,0,47,1
,47,0,47,1,47,0                           <DR>
1590 DATA▲0,0,0,45,0,60,1,60,0,60,1,60,1
,60,1,60,0,72,0                           <OV>
1600 DATA▲48,0,48,68,0,72,0,68,1,68,1,68
,0,53,1,53,0,53,1                         <TV>
1610 REM {CTL Q}{CTL R}{CTL R}{CTL R}2{C
TL R}{CTL R}{CTL R}                       <SV>
1620 DATA▲40,0,40,53,1,53,0,60,1,60,1,60
,0,45,0,47,0,60,1                         <BS>
1630 DATA▲40,0,40,60,1,60,1,60,1,60,1,60
,1,60,1,60,1,60,0                         <NP>
1650 DATA▲0,0,0,-1,0                      <FB>
30000 REM INIT                            <SG>
30010 DIM S$(64),B$(64),R$(16)            <ZW>
30020 R$="({CTL ,}{CTL F}{CTL ,}({CTL ,}
{CTL F}{CTL ,}{CTL F}{CTL F}{CTL Q}{CTL
Q}!!(("                                   <WI>
30030 B$="{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}nnnn????{CTL R
}{CTL R}{CTL R}{CTL R}0000{CTL ,}{CTL ,}
{CTL ,}{CTL ,}????nnnnnnnn????????{CTL ,
}{CTL ,}{CTL ,}{CTL ,}{CTL ,}{CTL ,}{CTL
 ,}{CTL ,}{CTL N}{CTL N}{CTL N}{CTL N}{C
TL N}{CTL N}{CTL N}{CTL N}"               <VO>
30040 S$="a0a0a0a0H$H${CTL B}?{CTL B}?a0
a0s:s:W+W+{CTL B}?{CTL B}?H$H$H$H${CTL B
}?{CTL B}?{CTL B}?{CTL B}?W+W+W+W+{CTL N
}{CTL N}{CTL N}{CTL N}{CTL N}{CTL N}{CTL
 N}{CTL N}"                               <KV>
30050 GRAPHICS 3+16:SETCOLOR 4,2,6        <NY>
30060 GRAPHICS 18:SETCOLOR 4,0,0:SETCOLO
R 0,0,15                                  <GW>
30070 ? #6:? #6;"▲▲M.C.S.C.▲********":? #
6;"▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲*▲▲*":? #6;:? #6;"▲SOFT
WARE▲▲****▲▲*"                            <LV>
30075 ? #6;"▲▲▲▲▲▲▲▲▲▲▲▲▲*▲▲****":? #6;"▲P
RESENT▲▲▲*▲▲*":? #6;"▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲*******" <KX>
30080 POSITION 0,9:? #6;"DIGI▲LOO▲/▲DIGI
▲LEY!"                                    <WY>
30085 ? #6;"▲(C)▲BY▲KEMAL▲EZCAN"          <ET>
30090 C=8:RETURN                          <VQ>
```

Listing 8. »Digi Loo-Digi Ley« (Schluß)

Aktion mit Action

Action zählt zu den schnellsten Programmiersprachen für Atari-Computer. Wer auf hohe Geschwindigkeiten Wert legt, aber nicht in Maschinensprache programmieren will, sollte Action in Erwägung ziehen.

In letzter Zeit machte eine Programmiersprache von sich reden, die ursprünglich als Sprache für Systementwickler konzipiert war: »C«. Sie wurde Anfang der siebziger Jahre von Dennis Ritchie aus BCPL weiterentwickelt. Als Standardwerk zur Programmierung in »C« wird daher auch oft das Buch von Kernighan/Ritchie (»The C Programming Language«) genannt. Ritchie war später auch an der Entstehung von UNIX beteiligt.

Es gab bereits mehrere Versuche, die Sprache C auf den Atari anzupassen: Das »Deep Blue C«-System ist weitgehend an einen klassischen C-Compiler angelehnt. Allerdings sind hier nicht nur beim Befehlsumfang, sondern auch bei der Arbeitsgeschwindigkeit Abstriche zu machen. Ein anderer Weg wurde mit C/65 begangen. Bei diesem System wird der Quelltext in einen Assemblertext compiliert, der dann mit MAC/65 (siehe Beitrag in diesem Sonderheft) weiter bearbeitet werden muß.

Unser Thema ist aber die Programmiersprache »Action«, die von OSS stammt. Diese Sprache enthält zwar viele Elemente von C, ist aber dennoch »anders« genug, um einen eigenen Namen verdient zu haben.

Action wird (womit wir schon beim ersten Unterschied zu konventionellen C-Compilern sind) auf einem Programmmodul geliefert. Wie bei allen anderen Cartridges von OSS handelt es sich dabei um ein 16 KByte-Super-Cartridge. Es belegt aber trotzdem nur 8 KByte-RAM-Speicher. Bei Verwendung von DOS XL kann der freie Speicherplatz sogar noch weiter erhöht werden. Der nächste Vorteil ist (genau wie bei MAC/65), daß sich Editor und Compiler gleichzeitig im Speicher befinden. Von der Konzeption des Systems her läßt sich Action also mit Turbo-Pascal vergleichen, bei dem ebenfalls Editor, Compiler und Quelltext gleichzeitig im Speicher vorliegen. Ein separater Linker ist nicht nötig.

Da Action, genau wie C, nicht mit Zeilennummern arbeitet, hat man dem Action-System einen stark an Textprogramme angelehnten Editor eingebaut.

Grundsätzlich stehen alle normalen Fähigkeiten des Bildschirmeditors zur Verfügung. Allerdings kann jede Zeile maximal 240 Zeichen lang sein, so daß man bei Schleifen und Kommentaren genügend Platz zum Einrücken hat. Da die Darstellung auf dem Bildschirm auf nur 40 Zeichen pro Zeile begrenzt ist (Bild 1), ist natürlich immer nur ein Teil des Listings zu sehen. Bei den Zeilen, die über den Rand des Bildschirms hinausreichen, ist jeweils das letzte Zeichen invertiert. Bewegt man nun den Cursor über den rechten Rand des Bildschirms, wird die Zeile, in der man sich befindet, nach links gescrollt (Bild 2). Weitere Fähigkeiten des Editors sind Funktionen zum Suchen und Ersetzen von Begriffen, der Einsatz eines zweiten Bildschirmfensters, in dem man eine andere Datei bearbeiten kann, und ähnliches mehr.

Das Fehlen von Zeilennummern beim Programmieren erschwert es oft, bestimmte Programmteile wiederzufinden: Der Action-Editor erlaubt aber, an beliebigen Stellen Markierungen (Tags) zu setzen, die man dann später mit nur zwei Tasten wieder anwählen kann.

Sämtliche Parameter wie Zeilenlänge und Fenstergröße etc. sind veränderbar.

Den Mittelpunkt des Action-Systems bildet der Monitor, von dem aus man Editor, Compiler und DOS aufrufen kann. Außerdem können einzelne Speicherzellen verändert und ganze Speicherbereiche gelistet werden. Wichtig beim Testen eines Programms ist der TRACE-Modus. Während des Programmablaufs erlaubt er die Ausgabe von Namen und Parametern jeder aufgerufenen Prozedur auf dem Bildschirm.

Die Programmiersprache Action ist weitgehend an C angelehnt. So haben beide grundsätzlich ähnliche Datentypen und eine ähnliche Syntax.

Da zwischen den einzelnen Befehlen nur Leerzeichen stehen brauchen, ist man in der Aufteilung des Quelltextes völlig frei.

Natürlich ist es empfehlenswert, bei Verschachtelungen, wie bei anderen Programmiersprachen auch, dazwischenliegende Zeilen einzurücken. So werden Programme übersichtlich und überschaubar.

An Datentypen sind in Action zunächst einmal BYTE und CHAR (einzelne Byte), INT (2-Byte-Integer zwischen -32768 und 32767) und CARD (desgleichen zwischen 0 und 65535) vertreten. Bei der Deklaration von Variablen kann man (muß aber nicht) angeben, welche Speicherstelle dazu benutzt werden soll. Beispiel: »BYTE chbase=$02F4«

Mit dieser Zeile würde man eine 1-Byte-Variable erzeugen, die genau auf der Adresse $02F4 liegt. Bei einer Wertzuweisung zu chbase wird die Adresse $02F4 geändert. Das Kommando »chbase=$CC« schaltet also den internationalen Zeichensatz ein. $02F4 fungiert nämlich als Basisregister für den Zeichensatz und $CC als High-Byte der Anfangsadresse des Zeichensatzes. Wo Basic umständliche POKE-Befehle erfordert, kann man hier sehr viel eleganter programmieren. Gleiches gilt natürlich für die anderen Datentypen.

Neben den fundamentalen Datentypen gibt es in Action auch erweiterte Arten von Variablentypen. Fangen wir mit dem Typ POINTER (Zeiger) an. Davon abgesehen, daß die Deklaration ein wenig anders abläuft (»CHAR POINTER pnt« statt »CHAR *pnt«), ist die Anwendung der Pointer in Action genauso wie in C gelöst. Mit »pnt=@var« erhält man die Adresse einer Variablen (in C:»pnt=&var«), mit »pnt ^« erhält man die Adresse, auf die pnt zeigt (in C müßte der Befehl dann »*pnt« lauten).

Ein weiterer wichtiger Datentyp ist das Feld (Array). Im Gegensatz zu C verwendet man in Action allerdings nur eindimensionale Arrays. Sie werden außerdem nur aus den fundamentalen Datentypen, also CHAR, BYTE, INT und CARD, aufgebaut. Allerdings kann man mit einem Trick auch Arrays aus den erweiterten Datentypen verwenden.

In Action lassen sich auch eigene Datentypen definieren; sie heißen Records. Man benutzt dazu die Anweisung TYPE, während es in C STRUCT heißt. Der Vorteil der Records liegt darin, verschiedene Datentypen unter einem gemeinsamen Oberbegriff zusammenfassen zu können.

Wie schon bei der Erörterung der fundamentalen Datentypen zu sehen war, erlaubt Action eine sehr maschinennahe Programmierung. Alle numerischen Konstanten können auch in hexadezimaler Schreibweise eingegeben werden, und auch sämtliche Funktionen zur Manipulation einzelner Bits sind verfügbar: »&« (logisches AND), »%« (OR), "!" (XOR), LSH (left shift) und RSH (right shift). Leider fehlt eine angenehme Eigenschaft von C. Es ist nicht möglich, Variablen an beliebigen Stellen im Programmtext durch zwei kurze Zeichen zu in- oder dekrementieren.

Immerhin kann man bei Veränderungen einer Variablen den Namen auf der rechten Seite des Gleichheitszeichens weglassen, vorausgesetzt man schreibt stattdessen zwei Gleichheitszeichen (Beispiel: »Zaehler==+1« erhöht die Variable Zaehler um eins).

Ein wichtiger Gesichtspunkt bei der Beurteilung einer Sprache sind die verfügbaren Strukturen für Schleifen und Verzweigungen. Action ähnelt in dieser Hinsicht weniger C, sondern eher Basic oder Pascal. Für Verzweigungen dient natürlich das Kommando IF. Hierbei ist die Anzahl der Befehle, die THEN folgen dürfen, völlig offen. Am Ende des durch THEN eingeleiteten Blocks muß allerdings ein FI stehen. ELSE ermöglicht es dann, den entgegengesetzten Fall abzufangen. Die Worte DO und OD beginnen beziehungsweise beenden einen Schleifenblock. Diese beiden Ausdrücke ersetzen die geschweiften Klammern, die üblicherweise zwischen einfachen Anführungszeichen eingebunden sind. Dabei kann man ohne Bedingungen arbeiten und erhält dann eine unendliche Schleife – oder aber WHILE, FOR oder UNTIL benutzen, um eine Bedingung für die Schleife festzulegen. Einen vorzeitigen Abbruch von Schleifen bewirkt der Befehl EXIT.

Was eine strukturierte Programmiersprache sein will, muß natürlich über Prozeduren, Funktionen und lokale Variablen verfügen. Während sich Prozeduren und Funktionen in C nur darin unterscheiden, daß Funktionswerte zurückgegeben werden (oder nicht), werden sie in Action bereits bei der Deklaration unterschieden.

Bei Prozeduren geht das folgendermaßen vor sich: Zunächst wird mit PROC angezeigt, daß eine Prozedurdeklaration stattfindet. Es folgen der Name der Prozedur und (in Klammern) Datentypen und Namen der übergebenen Parameter. Damit sind die übergebenen Parameter automatisch als lokale Variablen für diese Prozedur definiert.

Beendet wird der Vorgang mit dem Kommando RETURN. Prozeduren werden wie Kommandos aufgerufen (Beispiel: »Wait (30)«).

Funktionen unterscheiden sich von Prozeduren darin, daß sie einen Funktionswert an das aufrufende Programm zurückliefern können. Ein Aufruf einer Funktion sieht deshalb auch ein wenig anders aus. Beispiel: »X=STICK (0)«. Bei Funktionsdeklarationen muß zunächst der Typ des ermittelten Endresultats angegeben werden. Es folgt dann das Kommando FUNC, der Name der Funktion und die Liste der übertragenen Parameter (genau wie bei Prozeduren). Der einzige weitere Unterschied ist, daß zum Schluß dem Befehl RETURN noch ein Parameter übergeben wird, der dann als Funktionswert gilt.

Genau wie bei einem C-System werden auch zum Action-Compiler die wichtigsten Ein- und Ausgabefunktionen in Form einer Unterprogrammbibliothek mitgeliefert. Der große Unterschied ist allerdings, daß diese Routinen nicht, wie sonst üblich, auf Diskette beiliegen und dann in das compilierte Programm eingebunden werden, sondern im ROM des Action-Cartridges liegen. Dies hat zur Folge, daß compilierte Programme, wenn sie auf vordefinierte Funktionen oder Prozeduren zurückgreifen, nur mit eingestecktem Cartridge laufen.

Weiterhin gibt es, passend zum Action-Modul, eine Unterprogrammbibliothek. Darin enthalten sind zunächst Befehle zur Ausgabe sämtlicher Datentypen (inklusive vor PrintF). Daneben findet man, auch jeweils für alle Datentypen, den INPUT-Befehl. Ebenso sind alle anderen normalen Ein- und Ausgabefunktionen, wie GET, PUT, OPEN, CLOSE, XIO, NOTE und POINT, bereits als Prozedur definiert. Leider sucht man die recht wichtigen Befehle BGET und BPUT vergeblich. Grafik- und Soundbefehle, die zum größten Teil ihren Basic-Vorbildern entsprechen, fehlen aber nicht.

Da Action nicht ohne weiteres in der Lage ist, Zeichenketten zu verarbeiten, sind auch hierfür einige nützliche Prozeduren vordefiniert.

Wie wir bisher gesehen haben, hat Action zwar viele Eigenschaften von C, eine vollständige Implementation liegt aber beileibe nicht vor. Andererseits wartet Action mit Erweiterungen auf, die besonders auf dem Atari von sehr großem Nutzen sind. Dazu zählt zum Beispiel auch die Festlegung der Adresse einer Variablen.

Wer zeitkritische Programmteile doch lieber in Maschinensprache schreiben will, dem stehen zweierlei Wege offen. Für kurze, verschiebbare Programmteile empfiehlt es sich, den Maschinencode direkt als Datenblock in den Quelltext einzubinden. Längere Programme kann man ab einer festen Adresse assemblieren und die zugehörige Prozedur im Quelltext einzig und allein durch Festlegung ihrer Anfangsadresse definieren.

Für den professionellen Programmierer ist es wichtig zu wissen, daß er die volle Kontrolle darüber hat, wie der Compiler den Speicherplatz aufteilt. So läßt sich dann der erzeugte Code in ein EPROM brennen.

Weiterhin ist für den ernsthaften Einsatz wichtig, die Verwendung der im ROM integrierten Unterroutinen unterbinden zu können. Man definiert einfach die Routinen selbst neu. Einfacher ist es allerdings, auf die Action-Runtime-Library zurückzugreifen. Der Anwender bekommt eine Reihe vordefinierter Dateien, die mit dem INCLUDE-Befehl in die selbstverfaßten Programme eingebunden werden.

```
;      Erase8(127-x1+y0, 127+y1)
    FI
  FI
RETURN


PROC GetParam(STRING param, CARD ARRAY ▮
  CARD resultC
  STRING numBuf(0)=$550

  Print(param)
  Print(" = ")
  PrintC(cur^)
  IF initial THEN
    Print(", initial = ")
    PrintCE(initial^)
  ELSE
    PutE()
  FI

  Print("Enter new value:  ")
  resultC = InputC()
  IF numBuf(0)#0 THEN
ACTION! (C)1983 ACS
```

Bild 1. Eine Zeile kann bis zu 240 Zeichen lang sein. Da sich nur 40 Zeichen auf dem Bildschirm darstellen lassen, wird die Zeile, in der sich der Cursor befindet, gegebenenfalls nach links oder rechts gescrollt.

```
;      Erase8(127-x1+y0, 127+y1)
    FI
  FI
RETURN


▮STRING param, CARD ARRAY cur, initial)
  CARD resultC
  STRING numBuf(0)=$550

  Print(param)
  Print(" = ")
  PrintC(cur^)
  IF initial THEN
    Print(", initial = ")
    PrintCE(initial^)
  ELSE
    PutE()
  FI

  Print("Enter new value:  ")
  resultC = InputC()
  IF numBuf(0)#0 THEN
ACTION! (C)1983 ACS
```

Bild 2. Die mit dem Cursor gekennzeichnete Zeile ist bereits nach links gescrollt. Alle anderen Zeilen befinden sich an der ursprünglichen Position. So bleibt der zu bearbeitende Text übersichtlich.

Unter dem Strich gesehen handelt es sich bei Action (etwa 250 Mark) um eine professionelle Programmiersprache mit Arbeitsgeschwindigkeiten, die auf dem Atari bisher für eine höhere Programmiersprache als unerreichbar galten: Ein Benchmarkprogramm zur Berechnung der ersten 1000 Primzahlen war in Action nur 40 Prozent langsamer als die in Maschinensprache geschriebene Version.

Ein Grund für diese enorme Geschwindigkeit ist darin zu suchen, daß der Compiler intern völlig anders vorgeht als ein normaler C-Compiler. Durch diese vereinfachte maschinensprachnähere Arbeitsweise ist leider auch die Fähigkeit verlorengegangen, rekursiv zu arbeiten.

Als weiteren Zusatz zu Action bekommt man für etwa 100 Mark die »Programmer's Aid«-Diskette. Auf dieser Diskette findet man dann schmerzlich vermißte Befehle wie CIRCLE, Kommandos für Player/Missile-Grafik oder Fließkommaverarbeitung sowie einige Demoprogramme.

Action hat also für jeden etwas zu bieten: Dem Umsteiger von Basic wird eine schnelle, komfortable Sprache geboten, die auch dem Profi-Programmierer noch voll ausreicht. Und wer später plant, auf einem »großen« Computer in C weiterzumachen, kann sich bereits mit seinem Atari einarbeiten.

(Julian F. Reschke/wb)

Bezugsquellen:
CompyShop, Gneisenaustr. 29, 4330 Mühlheim/Ruhr, Tel. (0208) 497169
Münzenloher, Tölzer Str. 4, 8150 Holzkirchen, Tel. (08024) 1814

Das Textverarbeitungs-Sextett

Sechs verschiedene Textverarbeitungen auf dem Prüfstand. Welche ist die beste?

Den Heimcomputer als Schreibmaschine zu nutzen ist wohl eine der häufigsten ernsthaften Anwendungen im Heimbereich. Insgesamt sechs verschiedene Textverarbeitungen sollen deshalb verglichen werden, damit Sie beim Kauf die richtige Wahl treffen.

Eine der ältesten Textverarbeitungen für den Atari ist wohl der »Atari-Schreiber«. Er ist auf Modul und neuerdings auch auf Diskette lieferbar. Der »Atari-Schreiber« ist die deutsche Version des amerikanischen »Atari-Writers« und deshalb auch auf deutsche Verhältnisse abgestimmt. Sowohl das Handbuch wie auch die Benutzerführung im Programm sind in deutscher Sprache abgefaßt. Ebenso ist der Zeichensatz und sogar die Tastaturbelegung der deutschen Norm angepaßt. Wer sonst viel mit der Schreibmaschine umgeht, wird darüber erfreut sein. Alte Hacker auf dem Atari können sich allerdings anfangs wahrscheinlich vor Tippfehlern nicht mehr retten. Deshalb liegen dem Programm auch Aufkleber für die Tastatur bei, die die umbelegten Tasten entsprechend kennzeichnen.

Die Bedienungsanleitung zum »Atari-Schreiber« besteht aus drei Teilen. Das eigentliche Handbuch beinhaltet eine ausführliche Beschreibung des Programms und gibt Tips zum Umgang mit dem System. Zudem liegt eine Referenzkarte bei, in der alle Befehle von »Atari-Schreiber« kurz, aber informativ aufgeführt sind. Zum schnellen Nachschlagen von bestimmten Befehlen benutzt man am besten die Referenzkarte, die sich als nützliches Hilfsmittel beim Umgang mit der Textverarbeitung erweist.

Der »Atari-Schreiber« wird von einem Hauptmenü aus gesteuert, das die wesentlichen Funktionen zur Bearbeitung eines Textes enthält (Bild 1). Dieses Menü erscheint sofort nach dem Laden des Programms und erleichtert so den ersten Einstieg in die Textverarbeitung. Auch ohne lange im Handbuch nachschlagen zu müssen, kann man gleich mit der Eingabe des ersten Textes beginnen. Man betätigt einfach die Taste »N« und gelangt in den Schreib-Modus. Am oberen Bildschirmrand ist nun eine Zeile zu sehen, welche die grundlegenden Druckparameter, wie linke und rechte Randbegrenzung, oder den Zeilenabstand enthält. Am unteren Bildrand befindet sich die Anzeige über den aktuellen Cursorstand. Leider beziehen sich diese Angaben nur auf die Bildschirmposition und nicht auf die tatsächliche Position des laufenden Textes. So ist eigentlich nur die Spaltenangabe von Nutzen. Insgesamt lassen sich 21 Zeilen von je 35 Zeichen auf einmal auf dem Bildschirm darstellen. Ein horizontales Scrollen über die 35. Spalte hinaus findet nicht statt. Die Darstellung von Tabellen kann daher zum Problem werden. Die Formatierung eines Textes auf dem Bildschirm ist nicht identisch mit dem späteren Ausdruck. Der Text wird nur dem Bildschirmformat angepaßt; von den Einstellungen, die Ränder, Blocksatz und ähnliches betreffen, muß man sich leider erst auf dem Papier überraschen lassen.

Die Druckeranpassung ist für die Atari-Drucker 1020, 1025 und 1027 sowie für den Epson FX-80 über ein kleines Untermenü vorzunehmen. Andere Drucker werden vom Atari-Schreiber nicht unterstützt, so daß man in diesem Fall entweder für entsprechende Steuerzeichen im Text sorgen muß oder sich selbst ein passendes Drucksteuerprogramm schreibt.

Beim »Atari-Schreiber« ist die Ausgabe von Serienbriefen vorgesehen. Daher kann man einen Text auch mit Variablen versehen. Braucht man beispielsweise ein Rundschreiben mit gleichlautendem Text, aber persönlicher Anrede, so wird die Anrede durch eine Variable ersetzt. Beim Ausdruck des Rundschreibens schließlich steht anstatt der Variablen der eigentliche Text. Dieser wird einfach einer Datei von der Diskette entnommen, die die erforderlichen Daten enthält.

Stern oder Sternschnuppe?

Wer größeren Wert auf komfortable Benutzerführung legt, ist mit dem »Star-Texter« gut beraten. Allerdings meldet sich »StarTexter« mit lautem Getöse und grafischen Spielereien. So würde man eher ein neues Spiel an Stelle eines Textprogramms vermuten. Doch der erste Schein trügt. Wie schon der »Atari-Schreiber« ist auch dieses Programm auf den deutschen Benutzer zugeschnitten. Man verfügt also über einen deutschen Zeichensatz, kann aber auch beliebige andere Zeichensätze nachladen. Auf der Diskette ist deshalb ein spezielles Programm mit dem Namen »StarFont« enthalten. Damit kann man sich seine eigenen Zeichen definieren. Auf diese Art lassen sich natürlich auch spezielle grafische Symbole in einen Text aufnehmen, denn alle Sonderzeichen werden nicht nur auf dem Bildschirm dargestellt, sondern auch in dieser Form auf dem Drucker ausgegeben. Die Tastaturbelegung kann man sich aussuchen. Entweder arbeitet man mit der regulären Belegung der Tasten, oder man wandelt die Tastatur des Atari in eine deutsche Schreibmaschinentastatur nach DIN um, je nach eigenen Bedürfnissen und Gewohnheiten.

Nach dem Laden des Programms befindet man sich zuerst im normalen Textmodus. Die Darstellung des Textes erfolgt in 21 Zeilen mit je 40 Zeichen (Bild 2). Durch horizontales Scrollen können, je nach vorhergehender Randeinstellung, allerdings bis zu 80 Zeichen nebeneinander dargestellt werden. Die Spalte und Zeile, in der man sich befindet, entnimmt man der Anzeige am unteren Bildrand. Anders als beim »Atari-Schreiber« wird hier die tatsächliche Position im Text und nicht die Bildschirmposition angezeigt.

Drückt man die ESC-Taste, gelangt man in den sogenannten Control-Modus. Mit den Sondertasten START, OPTION und SELECT kann man eines der drei Menüs von »StarTexter« aufrufen. Das erste dieser Menüs bezieht sich rein auf Diskettenoperationen (Bild 3). Hier kann man beispielsweise Texte laden und speichern oder sich das Inhaltsverzeichnis einer Diskette ansehen. Daneben lassen sich auch eine Reihe von verschiedenen DOS-Operationen durchführen, wie das Löschen und Umbenennen von Files. Von hier aus lädt man auch andere Zeichensätze oder speichert die momentane Parametereinstellung auf Diskette.

Das zweite Menü ist so umfangreich, daß es gleich drei Bildschirmseiten in Anspruch nimmt. Hier werden alle Grundeinstellungen für die Textdarstellung vorgenommen, wie die Zeilenlänge oder der Blocksatz. Aber auch die farbliche Darstellung auf dem Bildschirm kann an die persönlichen Bedürfnisse angepaßt werden.

Das dritte Menü schließlich ist für den Ausdruck und das Formatieren von Texten zuständig. Die Anpassung an einen bestimmten Drucker erfolgt aber nicht hier, sondern bereits in einem eigenständigen Installationsprogramm. In diesem Basic-Programm, das unter dem Namen »INSTALL« auf Diskette abgelegt ist, wählt man entweder gängige Druckertypen direkt an, oder man definiert sich seine eigene Anpassung an den speziellen Drucker. Erwähnenswert ist hierbei, daß man nicht unbedingt über ein Centronics-Interface verfügen muß, um Texte an den Drucker zu schicken. Mit einem einfachen Kabel, dessen Herstellung im Handbuch beschrieben ist, kann man seinen Drucker auch an die Joystickports des Atari-Computers anschließen. Einzige Einschränkung des Druckbetriebes ist dabei jedoch, daß nicht alle Grafikzeichen des erweiterten Zeichensatzes ausgedruckt werden. Dies stört bei normaler Textverarbeitung aber kaum. Vor dem Drucken kann man sich den Text auch in 80-Zeichen-Darstellung betrachten, also in der Form, wie er später tatsächlich auf dem Papier erscheint. Diese Darstellung ist zwar aufgrund der mangelnden Auflösung des Atari nur schwer lesbar und nicht zum Editieren geeignet, vermittelt aber einen guten optischen Überblick über den fertigen Text.

Das Formatieren von Texten direkt auf dem Bildschirm ist mit »StarTexter« auch möglich. Jedoch ist dies bei umfangreicheren Texten eine sehr zeitraubende Angelegenheit. Hinzu kommt, daß aufgrund der Textformatierung recht verschwenderisch mit dem Speicherplatz, sowohl auf Diskette als auch im RAM-Speicher, umgegangen wird. Ein Beispiel: Begrenzt man den rechten Rand eines Textes auf Spalte 40, so wird trotzdem jede Zeile mit einer Länge von 80 Zeichen auf Diskette abgelegt. 40 Leerzeichen pro Zeile werden also umsonst gespeichert. Man benötigt folglich doppelt soviel Speicherplatz als eigentlich notwendig. Abhilfe schafft hier nur das ständige Umformatieren vor dem Speichern und nach dem Laden eines Textes, das aber, wie bereits erwähnt, recht lange dauert. Auch der Textspeicher leidet unter dieser Art der Darstellung. Maximal kann er 20000 Zeichen aufnehmen. Die Anzahl der Zeilen, die ein Text umfassen darf, ist jedoch auf 250 beschränkt. Schreibt man seinen Text mit einer Breite von 80 Spalten, so verfügt man auch tatsächlich über 80 mal 250 oder 20000 Zeichen Speicherplatz. Bei einer Randeinstellung bis zur Spalte 40 beispielsweise kann man nur mehr 40 mal 80, also 10000 Zeichen im Textspeicher unterbringen. Ideal scheint diese Lösung also nicht zu sein.

Textprogramm aus dem Alpenland

Das nächste Textverarbeitungsprogramm für den Atari nennt sich »Austro.Text«. Wie der Name schon vermuten läßt, stammt dieses Programm aus Österreich. Auch hier verfügt man neben einer deutschsprachigen Anleitung über einen deutschen Zeichensatz. Auf eine Umbelegung der Tastatur wurde jedoch bewußt verzichtet, um, laut Handbuch, Umgewöhnungsprobleme zu vermeiden. Ein Menü sucht man bei diesem Programm vergeblich. Alle Befehle zum Bearbeiten eines Textes muß man leider im Kopf haben beziehungsweise im Handbuch nachlesen. Bei der Befehlsausführung erscheint lediglich eine Art Statuszeile, in der man die Eingaben einträgt. So ist es gerade in der Einarbeitungsphase doch recht mühsam, mit den vielen verschiedenen Befehlen von »Austro.Text« zurechtzukommen. In der Praxis sieht das so aus: Will man beispielsweise den linken Rand des Textes beim Ausdruck auf Spalte 10 setzen, so fügt man das Kommando ».LM 10« in den Text ein. Entsprechend würde eine Begrenzung des rechten Randes auf Spalte 50 dann ».RM 50« lauten. Wie man sieht, sind diese Abkürzungen dem Englischen entlehnt (LM = Left Margin oder linker Rand), die sich – mit entsprechenden Englischkenntnissen – gut einprägen.

Bei diesem Programm wird ebenfalls kein separates Centronics-Interface zur Druckerausgabe benötigt. Mit einem speziellen Kabel, das man aber, anders als bei »StarTexter«, fertig kaufen muß, kann man einen Drucker mit Centronics-Schnittstelle auch über die Joystickports ansteuern. Zum Drucken eines Textes betätigt man gleichzeitig die START- und die P-Taste. Nun gibt man den Handlernamen (Gerätebezeichnung) ein, auf dem der Druck erfolgen soll. Der Handler »P:« spricht also die normale Centronics-Schnittstelle an, während »J:« den Ausdruck über die Joystickports veranlaßt. Auf diese Weise kann man einen Text bei entsprechender Wahl des Handlers natürlich auch als LIST-File auf Diskette oder Kassette ablegen und dann weiterverarbeiten. Spricht man den Handler »T:« an, so erscheint der Text auf dem Bildschirm im 80-Zeichen-Format. Die Darstellung entspricht dann genau dem Format, in dem der Text auf dem Drukker ausgegeben wird. Um Texte in diesem Modus darstellen zu können, muß man jedoch über mindestens 5500 Byte freien Speicherplatz verfügen. Wie auch in der normalen Textdarstellung (Bild 4) kann man übrigens hier den Feinscroll-Modus wählen, in dem der Text fließend über den Bildschirm rollt.

Zur Anpassung an die verschiedenen Druckertypen benutzt man auch hier ein eigenes Programm namens »Printer Editor«. Dieses Programm wird allerdings bereits beim Booten von »Austro.Text« in den Speicher geladen und belegt dann rund 7500 Byte Textspeicher. Das bedeutet, daß man noch 7500 Byte freien Speicher benötigt, um noch Änderungen zur Druckeranpassung vornehmen zu können. Ansonsten bleibt nur, den momentanen Text zu speichern und die Diskette neu zu booten.

A ohne Ä

»A-Text« ist als Textverarbeitungsprogramm für den Atari auch schon etwas länger auf dem Markt. Erfreulich ist die Tatsache, daß sich inzwischen das früher kritisierte Handbuch verbessert hat. Es erscheint nach wie vor in Deutsch, ist aber wesentlich übersicht-

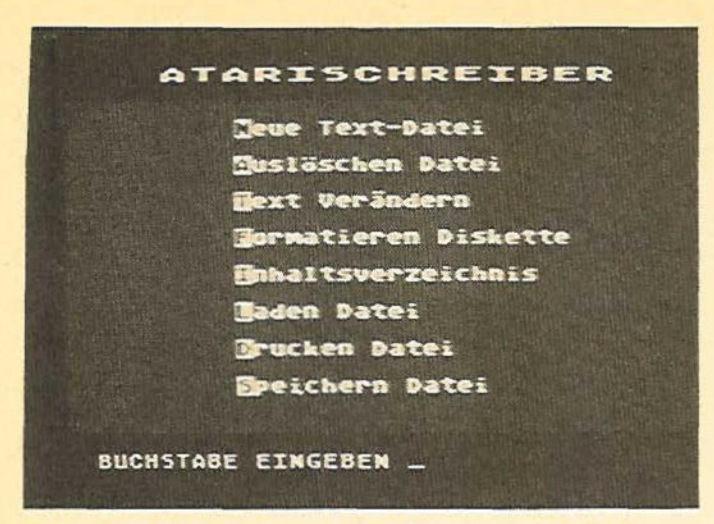


Bild 1. Das Hauptmenü vom »Atari-Schreiber« umfaßt alle wichtigen Funktionen.

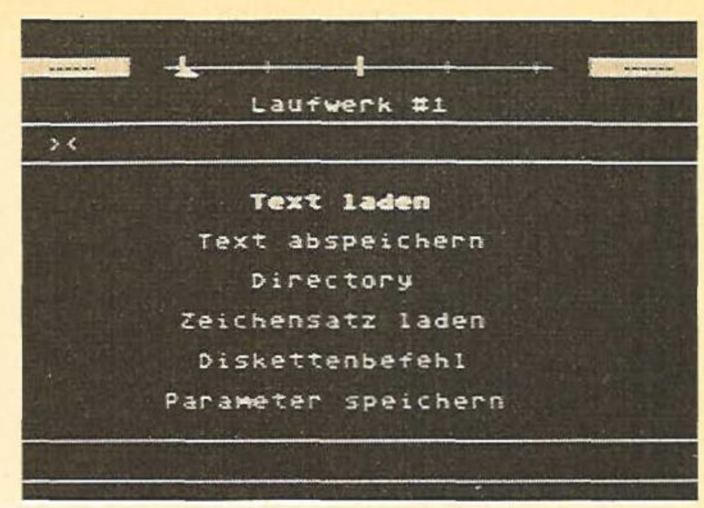


Bild 2. Von diesem Menü aus lassen sich beim »StarTexter« sämtliche Diskettenoperationen ausführen.

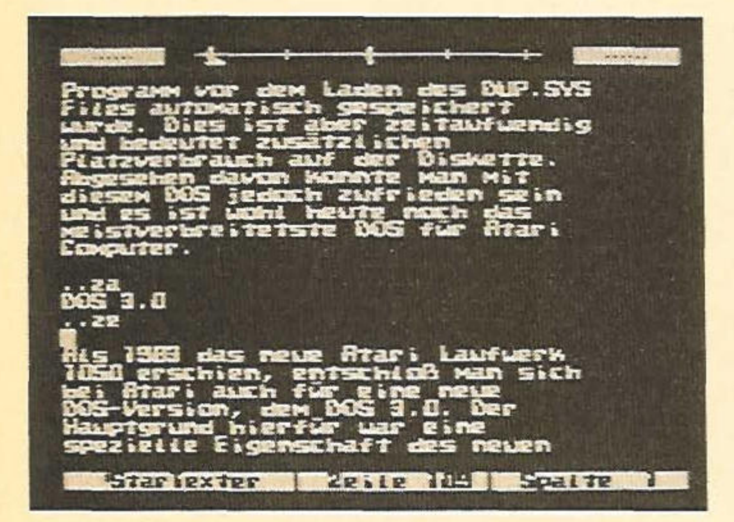


Bild 3. Die Textdarstellung von »StarTexter« mit geändertem Zeichensatz. Texte lassen sich, selbst mit anderem Zeichensatz, auf einem Drucker ausgeben.

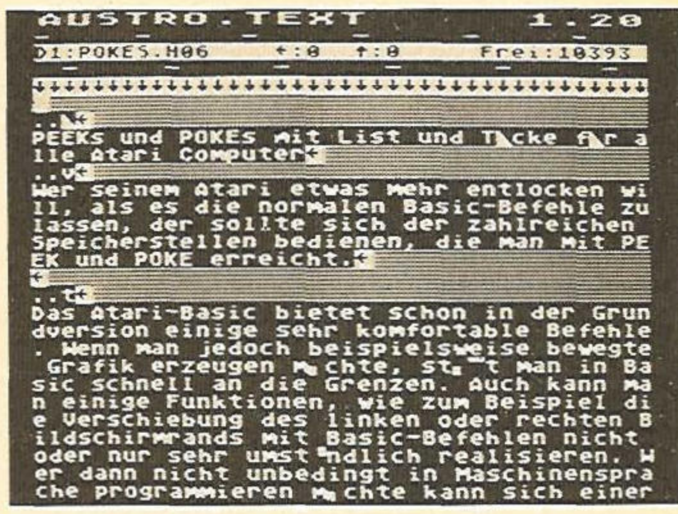


Bild 4. »Austro.Text« im Schreibmodus. Alle Leerzeichen und Leerzeilen werden mit Hilfe einer Schraffur optisch deutlich hervorgehoben.

licher geworden. Auf die Darstellung der deutschen Umlaute und Sonderzeichen auf dem Bildschirm muß man als »A-Text«-Benutzer jedoch verzichten. Statt dessen fügt man spezielle Grafikzeichen in den Text ein, die später bei der Druckerausgabe als Umlaute wiedergegeben werden.

Wie schon bei »Austro.Text« vermißt man bei diesem Programm die Menü-Steuerung. Die Kommandos müssen also auch hier im Kopf parat sein und dann in eine eigene Statuszeile eingegeben werden. Daß dies aber etwas langwierig werden kann, soll folgendes Beispiel aus dem Handbuch zeigen: Will man einen Text beispielsweise unterstreichen, so muß man folgende Sequenz eingeben:

»CTRL-L U CTRL-R CTRL-A CTRL-R ESC I RETURN«

Ähnlich sieht der Befehl zum Beenden des Unterstreichens aus. Beim »StarTexter« muß man vergleichsweise nur ein inverses A mit nachfolgender Null im Text einfügen, um denselben Effekt zu erzielen. Wie man sieht, ist die Handhabung von »A-Text« etwas gewöhnungsbedürftig.

Die Textdarstellung auf dem Bildschirm erfolgt in 20 Zeilen mit je 38 Zeichen. Ab der 20. Spalte beginnt der Bildschirm jedoch horizontal zu scrollen, so daß Zeilen mit einer maximalen Länge von 255 Zeichen ohne Zeilenvorschub eingegeben werden können. Vor allem bei Tabellen ist dies sehr nützlich und dient der Übersichtlichkeit.

Am oberen Bildrand sind ständig einige Informationen eingeblendet (Bild 5). So erfährt man die Zahl der bisher eingegebenen Zeichen, die Spalte, in der sich der Cursor befindet, und den restlichen Speicherplatz, der für den zu bearbeitenden Text zur Verfügung steht. Zudem ist die Größe des Kopierspeichers angegeben. Dieser Speicher umfaßt rund 4000 Zeichen und dient vor allem zur Verschiebung von Textblöcken. Am unteren Bildrand befindet sich die Statuszeile, die man über die ESC-Taste erreicht. Hier gibt man alle Kommandos ein, wie beispielsweise das Laden und Speichern von Texten.

Zur Anpassung des Programms an verschiedene Drucker existiert kein spezielles Programm. Alle Steuersequenzen für den Drucker müssen daher direkt im Text eingegeben werden. Laut Handbuch soll auch »A-Text« über die Joystickports mit einem Drucker in Verbindung treten können. Wie dies genau geschehen soll, bleibt selbst nach mehrmaligem Lesen des entsprechenden Kapitels unklar.

Perfektion aus den Staaten

Aus den USA kommt ein Textverarbeitungsprogramm mit dem Namen »Paperclip«. Entsprechend ist das ansonsten hervorragende Handbuch natürlich in Englisch abgefaßt. Auch ein deutscher Zeichensatz bleibt ein Wunschtraum. Sieht man hiervon ab, steht einem mit »Paperclip« allerdings ein Programm mit fast unbegrenzten Möglichkeiten zur Verfügung.

Bild 5. So kann bei dem Programm »A-Text« ein Text auf dem Bildschirm aussehen.

Bild 6. Das Arbeiten mit zwei Fenstern ist bei »Paperclip« ganz einfach und erleichtert so manche Problemstellung.

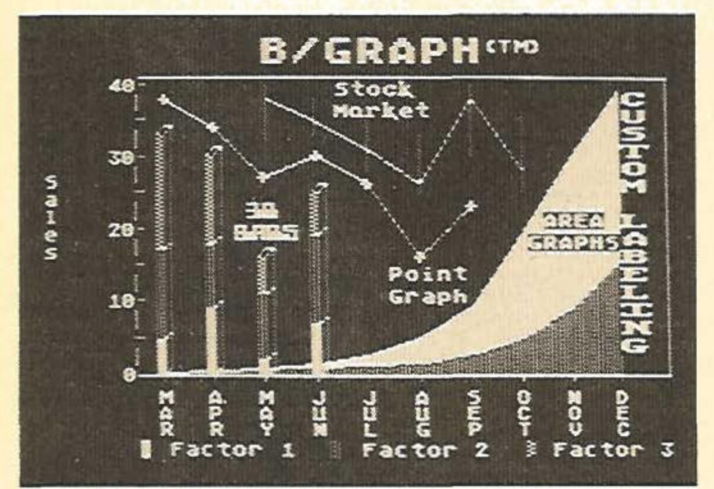


Bild 7. Eine solche grafische Darstellung läßt sich mit »Paperclip« zu Papier bringen.

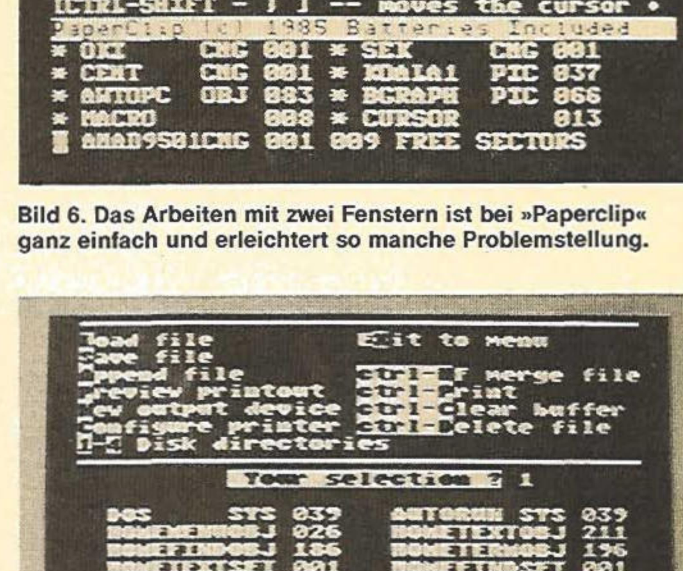


Bild 8. Ein Hauptmenü von »HomeText«. Jede Funktion läßt sich mit nur einer Taste aufrufen.

Zum Schutz vor Raubkopieren hat man sich hier übrigens etwas Besonderes einfallen lassen. Das Programm an sich ist nicht geschützt und soll sogar aus Sicherheitsgründen auf eine zweite Diskette kopiert werden. Zum Betrieb des Programms braucht man allerdings einen speziellen Schlüssel. Hierbei handelt es sich um eine kleine Box, die man am Joystickport 2 einstecken muß. Ohne diesen Schlüssel läuft gar nichts.

»Paperclip« ist zwar nur spärlich menügesteuert, jedoch fällt dies hier nicht besonders ins Gewicht. Erstens findet sich rasch jeder der einprägsamen Befehle in dem wirklich übersichtlichen Handbuch. Zweitens kann man bei Bedarf drei verschiedene Helpfiles aufrufen, in denen die Befehle kurz erläutert sind. Besonders praktisch ist hierbei, daß ein Helpfile in einem zweiten Fenster eingeblendet wird. Die wichtigsten Befehle können also parallel zum Text ständig eingesehen werden. Wer ohne Helpfile arbeitet, kann mit Hilfe der Fenster auch zwei verschiedene Texte gleichzeitig im Speicher bearbeiten (Bild 6). Das Hin- und Herschalten zwischen den Texten erledigt die SELECT-Taste.

Ist nur ein Fenster geöffnet, so wird der Text in 18 Zeilen mit je 39 Zeichen dargestellt. Die Zahl der Zeichen, die man mit dem horizontalen Scrolling nebeneinander schreiben kann, läßt sich zwischen 15 und 130 Zeichen einstellen. Wie der Text schließlich auf dem Drucker aussieht, kann man sich auch bei diesem Programm vorher ansehen. Die formatierte Darstellung erfolgt aber nicht mit 80 Zeichen Bildbreite, sondern in normaler Breite. Gegebenenfalls muß man den Text dann auf dem Bildschirm horizontal scrollen, um einen Gesamtüberblick zu erhalten.

Der Serienbrief-Ausdruck ist mit »Paperclip« kein Problem. Auch arbeitet dieses Programm ohne Schwierigkeiten mit bekannten Programmen wie »SynFile+« und »B/Graph« zusammen. Besonders interessant ist das ebenfalls auf der Diskette enthaltene Programm »HIRES«. Es handelt sich hierbei um ein Hardcopy-Programm, das Grafikbilder von unterschiedlichstem Format ausdrucken kann. Bild 7 zeigt hierzu ein Beispiel. Die grafische Darstellung einer Statistik, die man beispielsweise mit »B/Graph« erstellt hat, kann man ausdrucken und so in einen Text einbauen.

Der Druckeranpassung wurde bei diesem Programm besondere Aufmerksamkeit gewidmet. Auf der Diskette befinden sich bereits über 30 fertige Files, passend zu einer Vielzahl von Druckern. Sollte sich der eigene Drucker hier trotzdem noch nicht finden, existiert ein Anpassungsprogramm, in dem man die speziellen Druckeranforderungen festhalten kann.

Zum Ausdrucken von mehreren Files kann man sich übrigens ein sogenanntes Batch-File anlegen. In solch einem File führt man einfach alle zu druckenden Textfiles auf und speichert es dann. Beim Aufruf des Batch-Files werden alle darin vorkommenden Kommandos nacheinander ausgeführt. Diese Methode wird man vor allem dann anwenden, wenn ein Text in mehreren ein-

zelnen Teilen abgelegt ist und mit einem einzigen Befehl ausgedruckt werden soll.

100 Prozent Menü

»HomeText« ist eigentlich keine eigenständige Textverarbeitung, sondern Teil eines Programmpakets. Unter dem Namen »HomePak« erhält man nämlich neben einer Textverarbeitung auch ein Terminal- und ein Datenverarbeitungsprogramm.

Wie schon »Paperclip« kommt auch dieses Programm aus den USA. Sowohl das Handbuch wie auch die Benutzerführung sind also in Englisch ausgelegt. Dies stört bei »HomeText« jedoch kaum. Als einzige Textverarbeitung ist »HomeText« zu 100 Prozent menügesteuert. Allerdings ist auch hier kein deutscher Zeichensatz verfügbar, weder auf dem Bildschirm noch auf dem Drucker.

Die Stärken von »HomeText« liegen vor allem in seiner Kompaktheit und Übersichtlichkeit. Insgesamt kann man mit nur drei Menüs alle Funktionen dieses Programms ansprechen. Bild 8 zeigt eines der Menüs. Dabei versteht sich »HomeText« weniger als eigenständiges Textverarbeitungsprogramm, sondern ist vielmehr zur Zusammenarbeit mit den beiden anderen Programmen des Pakets konzipiert.

So kann man beispielsweise Daten des »Homefind«-Programms übernehmen und auf diese Weise Serienbriefe erstellen. »HomeTerm«, das DFÜ-Programm, holt sich Textdateien, die man per Akustikkoppler verdichten möchte, vom Textprogramm. Über DFÜ empfangene Texte kann man natürlich mit »HomeText« weiterverarbeiten. Alle drei Programme befinden sich auf nur einer Diskette und können durch ein Hauptmenü schnell und bequem aufgerufen werden. Auch beim Datenaustausch gibt es durch die aufeinander abgestimmten Programme keine Probleme.

Ein Nachteil von »HomeText« ist jedoch der etwas zu klein geratene Textspeicher. Er faßt nur 6620 Zeichen. Bei längeren Texten ist man also gezwungen, den Text in mehreren Files abzulegen. Die Textdarstellung erfolgt in 21 Zeilen mit 38 Spalten. Horizontales Scrollen über den rechten Bildrand hinaus ist nicht möglich. Dafür kann man sich mit dem Drucker-Preview ein Bild von dem fertigen Text machen. In skizzierter Form wird ein Blatt Papier auf dem Bildschirm dargestellt, auf dem der Text nur in Linien dargestellt ist. Dies genügt durchaus, um die äußere Form und Raumaufteilung des späteren Textes beurteilen zu können.

Die wichtigsten Daten der getesteten Textprogramme auf einen Blick

	Atari-Schreiber	StarTexter	AustroText	Paperclip	HomeText	A-Text
Anleitung in Deutsch	ja	ja	ja	nein (Englisch)	nein (Englisch)	ja
Datenträger	Modul oder Diskette	Diskette	Diskette	Diskette	Diskette	Modul, Diskette oder Kassette
Deutscher Zeichensatz	ja	ja	ja	nein	nein	nein
Zeichensatz wählbar	nein	ja	nein	nein	nein	nein
Tastatur umbelegt	ja	wahlweise	nein	nein	nein	nein
Printer Preview	nein	ja	ja	ja	grafisch angedeutet	nein
Horizontales Scrolling	nein	ja	nein	ja	nein	ja
max. Zeilenlänge	35	80	40	130	38	255
Druckeranpassung	teilweise voreingestellt	per Installations-programm	Installations-programm	Installations-programm	Installations-programm	mit Steuerzeichen im Text
Centronics-Interface notwendig	ja	nicht unbedingt	nicht unbedingt	ja	ja	nicht unbedingt
Grafik-Ausdruck	nein	Grafikzeichen	nein	Hardcopy-Programm für Bilder	nein	nein
Menügesteuert	ja	ja	Statuszeile	teilweise	ja	Statuszeile
Formatieren auf Bildschirm	nein	ja	nur in Preview	nur in Preview	nur in Preview	nein
Qualität des Handbuches	gut	befriedigend	befriedigend	sehr gut	gut, aber knapp	befriedigend
Feinscrolling	nein	nein	ja	nein	nein	nein
Blocksatz	*nur auf Drucker*	*auf dem Bildschirm und dem Drucker*	*auf dem Drucker oder in Preview*	*auf dem Drucker oder in Preview*	*nur auf dem Drucker*	*nur auf dem Drucker*
Umfang des Textspeichers	20448 Byte	20000 Byte, maximal 250 Zeilen	26762 Byte	keine Angabe	6620 Byte	28928 Byte
Serienbrief	ja	nein	nein	ja	ja	nein
Mathematische Funktionen	nein	ja	nein	ja	nein	nein
Preis	ca. 50 Mark	64 Mark	189 Mark	149 Mark	ca. 150 Mark	49 Mark

Kompromisse statt optimaler Lösungen

Für welches der sechs Textverarbeitungsprogramme man sich letztlich entscheidet, hängt vor allem davon ab, inwieweit man das Programm professionell nutzen möchte. Zumindest der deutsche Zeichensatz sollte sowohl auf dem Bildschirm als auch auf einem Drucker darstellbar sein. Eine gute Menüsteuerung wird man schnell schätzen lernen, wenn man nur ab und zu Texte bearbeitet und daher umfangreiche Kontrollcodes kaum im Gedächtnis behält. Kann man sich einen Text vor dem Drucken in seiner späteren Form ansehen, erspart dies so manches Blatt Papier. Vor allem bei Briefen, in denen es auch auf eine gewisse Raumaufteilung ankommt, trifft das zu. Sind häufig Tabellen zu erstellen, ist es von Vorteil, wenn die Zahl der Zeichen, die nebeneinander ohne Zeilenvorschub dargestellt werden können, möglichst groß ist. Zur professionellen Anwendung gehört natürlich auch der Serienbrief-Ausdruck. Doch wer sich die Vergleichstabelle ansieht, wird schnell feststellen, daß es das optimale Programm für den Atari nicht gibt. So bleibt dem Anwender immer nur ein Kompromiß und er wird auf das Programm zurückgreifen, das für ihn die wenigsten Nachteile aufweist.

(Wolfgang Czerny/wb)

Bezugsquellen:
CompyShop, Gneisenaustr. 29, 4330 Mülheim/Ruhr, Tel. (0208) 497169
Münzenloher, Tölzer Str. 4, 8150 Holzkirchen, Tel. (08024) 1814

150 Befehle mit Basic XE

Wem das Atari-Basic zu unkomfortabel und zu langsam ist, bekommt für etwa 300 Mark Basic XE auf Modul. Es ist das leistungsfähigste Basic für Atari-Computer mit bis zu 128 KByte-RAM.

Basic XE ist eine Weiterentwicklung von Basic XL, wobei natürlich den besonderen Fähigkeiten des neuen Atari-Computers 130XE Rechnung getragen wurde. Weiterentwicklung bedeutet vor allem die Hinzunahme neuer Befehle. Dabei sah man sich allerdings scheinbar zu Kompromißlösungen gezwungen. Kam man bei Basic XL noch mit einem Modul aus, so muß bei Basic XE neben dem Modul zusätzlich noch eine Diskette als Speichermedium herhalten. Dies birgt nun gleich zwei Nachteile in sich. Zum einen kann man ohne Diskettenlaufwerk nicht mehr alle Befehle von Basic XE nutzen, zum anderen belegt das File, das die zusätzlichen Befehle enthält, immerhin 91 Sektoren auf der Diskette. Dafür sind aber auf der Masterdiskette bereits einige kleine Programme enthalten, welche die Besonderheiten von Basic XE demonstrieren sollen. Neben Modul und Diskette erhält man natürlich auch ein Handbuch, das bedauerlicherweise in englischer Sprache gehalten ist. Doch auch ohne größere Englischkenntnisse sollten die Funktionen von Basic XE aufgrund der zahlreichen Beispiellistings in diesem Handbuch verständlich werden.

Zu den Befehlen, die Bestandteil des Diskettenfiles sind, gehört auch der Befehl »FAST«. Mit dieser Anweisung gelangt man in den sogenannten »Fast Modus« von Basic XE. Ist dieses Basic bereits im normalen Modus rund doppelt so schnell wie das Standard-Atari-Basic, so erhöht sich im »Fast Modus« die Verarbeitungsgeschwindigkeit von Programmen nochmals um den Faktor zwei. Die insgesamt also etwa vierfache Steigerung der Geschwindigkeit reicht so oftmals schon aus, um Programme zu schreiben, die sonst ohne Maschinensprache-Unterstützung zu langsam wären. Ebenfalls auf Diskette befindet sich ein File, das nur vom Atari 130XE oder aufgerüsteten Atari 800XL-Computern verarbeitet werden kann. Mit dem Befehl »EXTEND« spricht man die zusätzlichen 64 KByte an, die dem 130XE zur Verfügung stehen. Dieser Befehl verlegt das momentan im Speicher befindliche Programm in diese 64 KByte und schafft so Platz für andere Programme und Daten. Der Zusatzspeicher ist in vier Blocks von jeweils 16 KByte aufgeteilt. Jedem Block ist eine Nummer von 0 bis 3 zugeordnet. Ein spezieller POKE-Befehl erlaubt dann unter Angabe der Blocknummer, einzelne Daten in den Zusatzspeicher zu verlegen.

Player/Missiles mit Komfort

Zu den neuen Befehlen, die gegenüber Basic XL hinzugekommen sind, gehören auch eine Reihe von Anweisungen, die die Player/Missile-Programmierung betreffen. War man im normalen Basic noch gezwungen, eine Vielzahl verschiedener Adressen zu kennen, um Player oder Missiles anzusprechen, so erübrigt sich dies in Basic XE fast vollständig. Für nahezu jede Manipulation existiert ein eigener Befehl. »PMMOVE« beispielsweise veranlaßt eine beliebige Richtungsänderung eines Players oder Missiles. Auch die Kollisionsabfrage läßt sich mit einem einzigen Befehl mit dem Namen »BUMP« ausführen. Als recht nützlich erweist sich die Anweisung »PMCLEAR«, mit der sich der Speicherbereich, in dem ein Player oder eine Missile untergebracht ist, schnell und einfach löschen läßt.

Die Stringverarbeitung, die im normalen Basic etwas stiefmütterlich behandelt wird, ist nun wesentlich vereinfacht und damit auch komfortabler geworden. Es fängt damit an, daß man mit Basic XE auch indizierte Springfelder dimensionieren kann. Mit der Anweisung »DIM A$ (3,10)« beispielsweise werden drei Felder mit der Länge 10 erzeugt, die sich dann mit »A$(1)«, »A$(2)« und »A$(3)« aufrufen lassen. Einzelne Teile dieser Felder bearbeitet man dann mit den Befehlen »LEFT$«, »RIGHT$«, »MID$« oder »FIND$«. So gibt »FIND$« beispielsweise an, ob und wo sich einzelne Zeichen in einem String befinden. Das Sortieren von Arrays ist mit Basic XE ebenfalls kein Problem mehr. Hierzu verwendet man die Befehle »SORTUP« und »SORTDOWN«. Numerische oder alphanumerische Felder lassen sich also in aufsteigender oder abfallender Folge sortieren. Dabei kann man entweder das gesamte Feld oder nur ausgewählte Teilbereiche erfassen.

Programmieren, fast wie in Pascal

Auffallend sind vor allem auch Pascal-ähnliche Elemente, die in Basic XE auftauchen. Unterprogramme lassen sich zum Beispiel als Prozeduren gestalten. Diese können mit »CALL« aufgerufen und mit »EXIT« wieder verlassen werden. Innerhalb von Prozeduren kann man dann mit lokalen Variablen arbeiten. Das heißt, solche Variablen können zwar namensgleich mit anderen im Programm verwendeten Variablen sein, sind mit diesen jedoch nicht identisch. Ebenfalls von Pascal entlehnt ist sowohl die »WHILE-ENDWHILE«-Schleife als auch die bedingte Verzweigung »IF-ELSE-ENDIF«. Auf diese Weise lassen sich vor allem die uneleganten und zur Unübersichtlichkeit beitragenden »GOTO«-Anweisungen weitgehend aus einem Programm verbannen. Mit Basic XE ist es also möglich, bis zu einem gewissen Grad strukturiert zu programmieren und so eine wesentlich bessere Übersichtlichkeit eines Programms zu erzielen. Dies wird zudem durch die Tatsache unterstützt, daß ein Listing in Basic XE formatiert ausgegeben wird.

```
100 Rem -------------------
110 Rem BEISPIEL-LISTING
120 Rem mit Basic XE
130 Rem -------------------
140 Graphics 0
150 ? "RATEN SIE EINE ZAHL"
160 ? "ZWISCHEN 1 UND 100"
170 ?
180 Zahl=Random(1,100)
190 While Versuch<>Zahl
200   ? "WIE LAUTET IHR VERSUCH ?";
210   Input Versuch
220   If Versuch<Zahl
230     ? "DIE ZAHL IST ZU KLEIN"
240   Else
250     If Versuch>Zahl
260       ? "DIE ZAHL IST ZU GROSS"
270     Endif
280   Endif
290 Endwhile
300 ? :? "GEFUNDEN"
310 End
```

Listing. Ein typisches Basic XE-Programm.

Schleifen und »IF«-Anweisungen werden also automatisch eingerückt und so optisch hervorgehoben. Ein typisches Basic XE-Programm zeigt das Listing 1.

Viele der neuen Routinen sind dem geübten Programmierer bereits von oft benötigten Maschinensprache-Unterprogrammen her bekannt. So verschiebt man Speicherblöcke mit dem Befehl »MOVE«. Dadurch kann man beispielsweise den Inhalt des Bildschirmspeichers in einem String ablegen oder den Atari-Zeichensatz kopieren. Die automatische Zeilennumerierung bei der Programmeingabe und die Neunumerierung eines bestehenden Programms sind ebenfalls ohne Maschi-

nenhilfsroutinen durchführbar. Hierzu dienen die Befehle »NUM« und »RENUM«. Benötigt man eine Liste aller im Programm verwendeten Variablen, so erlaubt das der Befehl »LVAR«. Zur schnellen Umwandlung von dezimalen in hexadezimale Zahlen benutzt man den »HEX$«-Befehl. Nützlich ist der Befehl »SET«. Mit ihm lassen sich eine Reihe von grundsätzlichen Parametern allgemeiner Art ändern. So ermöglicht dieser Befehl beispielsweise die Auswahl, mit welchem Zeichen eine Input-Anweisung den Benutzer zur Eingabe auffordert. Die Funktion der »BREAK«-Taste kann auch auf diese Weise unterbunden werden. Selbst das Dimensionieren von Strings erübrigt sich mitunter. Bis zu einer Größe von 255 Zeichen, abhängig von der jeweiligen »SET«-Einstellung, werden Strings nämlich automatisch dimensioniert. Insgesamt können 16 verschiedene Parameter mit diesem Befehl beeinflußt werden. Die Einstellung der Parameter fragt man mit dem Befehl »SYS« ab.

Hilfreich bei der Fehlersuche ist der »TRACE«-Befehl. Vor der Ausführung einer Programmzeile erscheint die jeweilige Zeilennummer auf dem Bildschirm. So kann man leicht kontrollieren, ob sich ein Programm in den gewünschten Bahnen bewegt. Die Textausgabe erfolgt wahlweise invers oder normal, was einfach durch die beiden Befehle »INVERSE« und »NORMAL« gesteuert wird. Man kann einen auszugebenden Text aber auch noch formatiert darstellen. Dies geschieht mit dem »PRINT USING«-Befehl.

Auch einige DOS-Funktionen finden sich in Basic XE wieder. So läßt sich das Inhaltsverzeichnis einer Diskette zum Beispiel mit dem Befehl »DIR« abrufen. Für das Löschen und Umbenennen von Files auf Diskette existieren eigene Befehle. Neben den gewohnten Befehlen »GET«, »PUT«, »PRINT« und »INPUT«, die zum Lesen und Schreiben von Daten auf Diskette verwendet werden, stellt Basic XE zu diesem Zweck einige neue Befehle zur Verfügung. Sie zeichnen sich vor allem durch eine höhere Ausführungsgeschwindigkeit aus. Mit »BPUT« und »BGET« schreibt beziehungsweise liest man ganze Speicherblöcke von Diskette. Auf diese Weise lassen sich beispielsweise Grafikbilder schnell und einfach von Diskette laden oder speichern. »RPUT« und »RGET« sind für das Speichern und Lesen von Records mit fester Länge verantwortlich. Dabei kann ein Record sowohl aus einem String wie auch einer Gleitkommazahl bestehen. Schließlich gibt es noch die Befehle »BLOAD« und »BSAVE«, die man für die Behandlung von Binärfiles im »DOS LOAD«-Format benutzt. Normale Maschinensprachefiles können also von Basic aus geladen und gestartet werden.

Gut, aber teuer

Die Qualität der Befehle und der Befehlsumfang (Basic XE umfaßt zirka 150 Befehle) lassen kaum Wünsche offen. Lediglich das Laden zusätzlicher Befehle von Diskette scheint keine glückliche Lösung zu sein. Das entscheidende Kaufhindernis dürfte allerdings der Preis von Basic XE sein. Mit etwa 300 Mark ist es wohl eines der teuersten Programme, die derzeit für Atari-Computer angeboten werden. Der Preis fällt um so mehr ins Gewicht, wenn man bedenkt, daß beispielsweise Turbo-Basic XL bei ähnlichen Leistungsmerkmalen doch wesentlich billiger zu haben ist.

(Wolfgang Czerny/wb)

Bezugsquelle:
CompyShop, Gneisenaustr. 29, 4330 Mühlheim/Ruhr,
Tel. (0208) 497169

DOS-Parade

Wichtigstes Hilfsmittel zum Arbeiten mit einer Diskettenstation ist das DOS. Doch welche der verschiedenen DOS-Versionen soll man wählen?

DOS ist die Abkürzung für Disk Operating System oder auf deutsch Disketten Betriebssystem. Betreibt man ein oder mehrere Diskettenlaufwerke am Atari, so kommt man ohne ein solches DOS nicht aus. Die zum Betrieb eines Laufwerks nötigen Routinen sind nämlich nicht im eigentlichen Betriebssystem des Atari verankert, sondern müssen stets von Diskette geladen werden. Dies hat den Vorteil, daß man sich, abhängig vom verwendeten Laufwerk und den speziellen Bedürfnissen, das jeweils optimale DOS heraussuchen kann.

Von Atari selbst gibt es nun schon die vierte offizielle DOS-Version. Angefangen hat es mit »DOS 1.0«. Diese erste Version, die zusammen mit dem Atari-Laufwerk 810 Ende 1979 erschien, war noch mit etlichen Nachteilen und kleineren Fehlern behaftet. So gab es beispielsweise noch kein »AUTORUN.SYS«-File, um Programme nach dem Einschalten des Computers automatisch zu laden und zu starten. Andererseits konnte zum Kopieren von Programmen und Disketten nur ein relativ kleiner Puffer und nicht der gesamte RAM-Speicher verwendet werden. Auch war der wahlfreie Zugriff auf einzelne Byte in bestimmten Sektoren mit den NOTE- und POINT-Befehlen noch nicht implementiert. Es verwundert also kaum, daß Atari bereits Anfang 1980 eine neue Version, nämlich »DOS 2.0« (Bild 1) herausbrachte.

Dieses DOS war nun frei von den Fehlern und den gröbsten Nachteilen der ersten Version. Allerdings ging man auch hier Kompromisse ein. War DOS 1.0 noch ständig komplett im Speicher vertreten, teilte man DOS 2.0 in zwei getrennte Files auf. Zum einen in das DOS.SYS-File, das durch den Boot-Vorgang in den Speicher des Atari gelangte, zum anderen in das DUP.SYS-File, das nur bei Bedarf durch den DOS-Aufruf von Basic aus geladen wurde. Vorteil dieser Methode ist die Einsparung von wertvollem Speicherplatz. Nur noch rund 2 Kilobyte des RAM-Speichers gingen durch die Belegung des DOS.SYS-Files verloren. Die wichtigsten Funktionen, wie das Speichern und Laden von Programmen waren dabei noch durchführbar. Auch konnte man mit speziellen XIO-Routinen beispielsweise Disketten formatieren oder Files auf Diskette löschen. Spätestens aber zum Kopieren von Programmen oder zur Ausgabe des Disketteninhaltes auf dem Bildschirm, mußte man das DUP.SYS-File aufrufen. Dabei wurden im Arbeitsspeicher befindliche Basic-Programme jedoch überschrieben. Zur Rettung des Speicherinhaltes mußte man sein Programm daher entweder vor jedem DOS-Aufruf speichern oder aber ein sogenanntes MEM.SAV-File anlegen, in welches das Programm vor dem Laden des DUP.SYS-Files automatisch gespeichert wurde. Dies ist aber zeitaufwendig und bedeutet zusätzlichen Platzverbrauch auf der Diskette. Abgesehen davon konnte man mit diesem DOS jedoch zufrieden sein, und es ist wohl heute noch das meistverbreitete DOS für Atari-Computer.

Als 1983 ein verbessertes Atari-Laufwerk 1050 erschien, entschloß man sich bei Atari auch für eine neue DOS-Version, dem DOS 3.0 (Bild 2 zeigt das Hauptmenü von DOS 3.0).

```
DISK OPERATING SYSTEM II VERSION 2.0S
COPYRIGHT 1980 ATARI

A. DISK DIRECTORY  I. FORMAT DISK
B. RUN CARTRIDGE   J. DUPLICATE DISK
C. COPY FILE       K. BINARY SAVE
D. DELETE FILE(S)  L. BINARY LOAD
E. RENAME FILE     M. RUN AT ADDRESS
F. LOCK FILE       N. CREATE MEM.SAV
G. UNLOCK FILE     O. DUPLICATE FILE
H. WRITE DOS FILES

SELECT ITEM OR RETURN FOR MENU
```

Bild 1. Das Hauptmenü zu DOS 2.0. Sämtliche Disketten-Operationen lassen sich von hier ausführen. Da Disketten allerdings mit maximal 720 Sektoren zu formatieren sind, ergibt sich eine Gesamtspeicherkapazität von rund 88 KByte.

```
Atari DOS 3                    Copyright 1983

Inhalt          Laden          Mem sav
Zum Modul       Speichern      Geh nach
Kopieren        Vernichten      hex addr
Duplizieren     Neunennen      X-Funktion
Formatieren     Protect
Wandle DOS 2    Unprotect      Hilfen

Deine Wahl? Inhalt
Dateispezifikation? D1:*.*
Ausgabegeraet? E:
```

Bild 2. DOS 3.0 präsentiert sich mit einem etwas anders aufgebauten Menü. Als Besonderheit gibt es hier die »Hilfen«-Funktion, mit der man sich erklärende Texte zu den einzelnen Menüfunktionen direkt auf den Bildschirm holen kann.

Der Hauptgrund hierfür war eine spezielle Eigenschaft des neuen Laufwerks. Neben der normalen Schreibdichte, die auch das Laufwerk 810 beherrscht, kann das 1050er Laufwerk in einer höheren Schreibdichte (enhanced density) arbeiten. Dies ist nicht mit der doppelten Schreibdichte (double densitiy) zu verwechseln. Denn dies würde beim Atari 1050-Laufwerk bedeuten, daß pro Sektor statt 128 Byte 256 Byte Platz fänden. Bei gleicher Sektorenzahl würde man also die doppelte Speicherkapazität erreichen.

DOS 3.0

Erhöhte Schreibdichte bedeutet nur, daß statt 18 jetzt 26 Sektoren pro Spur angelegt werden können. Die Anzahl erhöht sich also von 720 auf 1040. Praktisch verfügt man nur über 1023 Sektoren, da das Laufwerk die übrigen 17 nicht mehr verwalten kann.

Bei DOS 3.0 kam man auf die kuriose Idee, jeweils acht Sektoren zu einem Block von 1024 Byte zusammenzufassen. Dies erleichtert dem Directory zwar die Verwaltungsarbeit, hat aber zwei entscheidende Nachteile. Einmal wird dadurch in der Regel enorm viel Platz verschwendet: Die Mindestlänge eines Programms auf der Diskette beträgt nun nämlich einen Block beziehungsweise acht Sektoren. Selbst ein Programm mit einer Länge von nur 10 Byte belegt damit einen ganzen Block, also 1024 Byte. Außerdem ist dieses Format unglücklicherweise nicht kompatibel zu DOS 2.0. Eine mit DOS 3.0 formatierte Diskette kann also nicht unter DOS 2.0 gelesen werden. Umgekehrt sind die alten Atari-Laufwerke 810 nicht mehr in der Lage, DOS-3.0 Disketten zu verarbeiten. Ein unter DOS 2.0 erstelltes File kann zwar auf DOS 3.0-Format umgewandelt werden, umgekehrt ist dies jedoch nicht durchführbar. Zum Trost sei aber gesagt, daß zu diesem Zweck bereits Utilities in verschiedenen Zeitschriften erschienen sind.

Trotz dieser Nachteile mußte man bis zur Einführung des neuen Atari 130XE warten, bis die vorläufig letzte Version, nämlich DOS 2.5, erschien. DOS 2.5 kann man getrost als das beste DOS bezeichnen, das von Atari bisher geliefert wurde (Bild 3). Es unterstützt die erhöhte Schreibdichte des 1050er Laufwerks. Die Diskettenverwaltung durch Blocks ist aufgehoben worden, so daß die Filekompatibilität zur Version 2.0 wieder hergestellt ist. Es sind also einzelne Sektoren angesprochen, deren Anzahl aber gestiegen ist. Dies hat auch den Vorteil, daß man Disketten mit erhöhter Schreibdichte wenigstens zum Teil mit dem alten Laufwerk 810 lesen kann. Lediglich die Sektoren zwischen 720 und 1023 bleiben dem 1050er Laufwerk vorbehalten. Sind mehr als 999 Sektoren frei, so werden diese übrigens nicht mehr angezeigt. Dadurch sollte man sich jedoch nicht irritieren lassen. Natürlich unterstützt DOS 2.5 auch die 128 KByte des 130XE. Vor allem kann man im zusätzlichen – 64 KByte umfassenden – RAM-Speicher eine sogenannte RAM-Disk anlegen, die wie ein Diskettenlaufwerk mit der Laufwerknummer 8 angesprochen wird. Man verfügt dann über rund 500 »Sektoren«, deren Inhalt quasi ohne Zeitverlust geladen oder beschrieben werden kann. Außerdem hat man noch einige andere neue Fähigkeiten in das DOS aufgenommen, die zwar nicht im Menü aufgeführt sind, jedoch über ».COM«-Files ausgeführt werden können. Versehentlich gelöschte Programme wiederherzustellen, sofern sie nicht bereits überschrieben wurden, ist nun kein Problem mehr. DOS 3.0-Files lassen sich in das DOS 2.5-Format umwandeln. Auch kann man AUTORUN.SYS-Files zum automatischen Programmstart von Basic-Programmen

```
DISK OPERATING SYSTEM II VERSION 2.5
COPYRIGHT 1984 ATARI CORP.

A. DISK DIRECTORY  I. FORMAT DISK
B. RUN CARTRIDGE   J. DUPLICATE DISK
C. COPY FILE       K. BINARY SAVE
D. DELETE FILE(S)  L. BINARY LOAD
E. RENAME FILE     M. RUN AT ADDRESS
F. LOCK FILE       N. CREATE MEM.SAV
G. UNLOCK FILE     O. DUPLICATE FILE
H. WRITE DOS FILES P. FORMAT SINGLE

SELECT ITEM OR RETURN FOR MENU
```

Bild 3. Wer DOS 2.0 bereits kennt, kommt mit DOS 2.5 schnell zurecht. Der einzige Unterschied: Mit »I« formatiert man eine Diskette im enhanced Modus und mit »P« erhält man ein Diskettenformat identisch mit DOS 2.0.

```
OSS DOS XL - ATARI version 2.30
 Copyright (C) 1983   OSS,Inc.

D1:DIR
* DOS      SYS 046
* DOSXL    SUP 046
* DOSXL    XL  059
* MENU     COM 025
* BUG65    COM 074
* CONFIG   COM 036
* COPY     COM 075
* DO       COM 003
* DUPDBL   COM 011
* DUPDSK   COM 011
* INIT     COM 006
* INITDBL  COM 023
* NOVERIFYCOM 001
* RS232    COM 001
* RS232FIXCOM 002
* IOMAC    LI
```

Bild 4. DOS XL befindet sich ständig im RAM-Speicher. Gibt man von Basic aus »DOS« ein, gelangt man ohne Diskettenzugriffe zur Kommandozeile. Befindet sich zu dem Zeitpunkt ein Basic-Programm im Speicher, wird es nicht gelöscht.

```
Happy-Computer DOS II+/D      V:4.5M
(c) Stefan Dorndorf

D1:DIR
  TURBOXL  COM 145
  COMPILERCOM 080
  RUNTIME  OBJ 088
  STRATEGOBAS 187
  APFEL    FRO 063
  DOS      SYS 037
  MASTER   COM 052
0055 FREE SECTORS.

D1:COP APFEL.FRO
-> Destination
```

Bild 5. Happy-DOS (Listing des Monats in Ausgabe 3/86) ist ähnlich aufgebaut wie DOS XL. Es liegt ebenfalls stets im Speicher vor. Als besonderen Leckerbissen bietet Happy-DOS noch eine RAM-Disk, die insgesamt 12 KByte umfaßt.

erzeugen. Da für jedes vorgesehene Laufwerk 128 Byte Speicherplatz im DOS reserviert werden, ist es ein leichtes, die Anzahl der verwendeten Laufwerke an die eigene Konfiguration anzupassen. So lassen sich auch noch einige Byte einsparen. Etwas störend wirkt sich die Anordnung des Directory-Eintrags auf einer DOS 2.5 formatierten Diskette aus. Da dieser Eintrag nicht in einem Stück, sondern verteilt auf der Diskette plaziert ist, macht sich der Schreib-/Lesekopf des Laufwerks immer wieder durch lautstarke Positionswechsel bemerkbar.

Natürlich gibt es auch DOS-Versionen von anderen Herstellern als Atari, beispielsweise DOS XL 2.3 von OSS. Unter dieser DOS-Version laufen so bekannte Programme wie »Action«, »MAC/65« oder »Basic XL«. Obwohl dies der Name nicht vermuten läßt, ist DOS XL für alle Atari-Computer verwendbar, da drei verschiedene Versionen auf der Master-Diskette vorhanden sind – passend zu der jeweiligen Speicherkonfiguration. Die Kommandozeile mit einem teilweise aufgelisteten Directory zeigt Bild 4. Neben der normalen Schreibdichte wird bei DOS XL sogar die doppelte Schreibdichte unterstützt. Um dies zu nutzen, muß man allerdings entweder das Atari 1050-Laufwerk hardwaremäßig erweitern (wie im Artikel »Floppy-Speeder« in diesem Sonderheft beschrieben), oder man verwendet ein Laufwerk eines Fremdherstellers. Die Speicherkapazität einer Diskette steigt dann allerdings von 90 KByte unter normaler Schreibdichte oder 130 KByte bei erhöhter Schreibdichte auf stolze 176 KByte. Dieses Format ist dann natürlich von den Atari-DOS-Versionen nicht mehr lesbar. Verzichtet man aber auf doppelte Schreibdichte, so besteht Filekompatibilität zwischen DOS XL und DOS 2.0 beziehungsweise DOS 2.5.

DOS XL verzichtet auf eine Trennung zwischen DOS.SYS und DUP.SYS-File. Das heißt aber nicht, daß keine weiteren Files nachgeladen werden. Man unterscheidet hier zwischen Kommandos, die das DOS ohne weiteres Zutun ausführt und Anweisungen, die nur durch Nachladen eines entsprechenden Files ausgeführt werden. Dies hat den Vorteil, daß das eigentliche DOS nur wenig Speicherplatz im RAM belegt, zumal es recht geschickt hinter dem Betriebssystem ROM abgelegt ist. Im eigentlichen Basic-RAM-Bereich befindet sich nur noch eine Sprungtabelle, die ungefähr 500 Byte beansprucht. Der Nachteil dieser Methode besteht jedoch darin, daß die extern gelagerten Files relativ viel Platz auf der Diskette verbrauchen. Es ist also ratsam, auf die jeweilige Arbeitsdiskette nur diejenigen DOS-Files zu kopieren, die man unbedingt benötigt. Auch das Menü von DOS XL gehört zu diesen Files, denn im Normalfall arbeitet DOS XL nur mit einer Kommandozeile. Hat man alle DOS-Befehle im Kopf, läßt es sich auf diese Art sogar wesentlich schneller arbeiten als mit einem Menü.

Eine Besonderheit stellt die sogenannte Batch-Verarbeitung dar. Man kann hierbei beliebigen Text und DOS-Anweisungen in einem File zusammenstellen. Versehen mit dem Extender ».EXC« besitzt man dann ein File, das man von der Kommandozeile aus aufrufen kann. Das File »STARTUP.EXC«, das dem herkömmlichen »AUTORUN.SYS« entspricht, ist ein Beispiel für so ein Batch-File. Auch häufig verwendete Befehlssequenzen lassen sich auf diese Art zeitsparend ausführen. Neben den kommerziellen Programmen kann man natürlich auch auf die verschiedensten DOS-Versionen zurückgreifen, die beispielsweise als Public Domain-Programme erhältlich sind, oder als Listings in Zeitschriften veröffentlicht werden. Ein Beispiel ist das in Happy-Computer, Ausgabe 3/86 als Listing des Monats abgedruckte Happy DOS (Bild 5). Ähnlich wie bei DOS XL wird nur mit einer Kommandozeile gearbeitet. Jedoch sind alle Befehle im DOS.SYS-File enthalten, so daß ein Nachladen von Files entfällt. Trotzdem beansprucht das DOS-File nur 37 Sektoren auf Diskette. Zusätzlich ist noch eine 12 KByte große RAM-Disk integriert, die auch mit den normalen Atari-Computern angesprochen werden kann. Es läßt auch zu, wie in DOS 2.5, Disketten mit erhöhter Schreibdichte zu formatieren.

Die Entscheidung für ein bestimmtes DOS sollte vor allem von der allgemeinen Filekompatibilität abhängig gemacht werden. Speziell DOS 3.0 sollte man trotz seiner relativ guten Benutzerführung also vermeiden. Kommt es auf äußerste Platzersparnis im RAM an, oder besitzt man ein Laufwerk für doppelte Schreibdichte, so ist man mit DOS XL am besten bedient. Allgemein kann man jedoch sagen, daß DOS 2.5 von Atari das derzeit interessanteste Disketten-Betriebssystem darstellt. Arbeitet man mit Basic-Versionen wie zum Beispiel Turbo-Basic XL, kann man auf den Sprung ins DOS-Menü und somit auf das Laden der DUP.SYS-Datei auch meist verzichten. Einige DOS-Funktionen sind dann nämlich bereits als Basic-Befehle vorhanden. Die integrierten Utilities und die Diskettenformatierung in normaler oder erhöhter Schreibdichte, lassen also gute Zukunftsaussichten für DOS 2.5 vermuten.

(Wolfgang Czerny/ub)

Bezugsquelle: CompyShop, Gneisenaustr. 29, 4330 Mühlheim/Ruhr, Tel. (0208) 497169

MAC/65: ein Assembler sprintet los!

Schnell stößt man in Basic an die Grenzen der erreichbaren Geschwindigkeit. Selbst eine optimale Programmierung verleiht Basic keine Flügel. Wer es eilig hat, sollte auf einen Assembler zurückgreifen.

Es gibt sicherlich verschiedene Gründe dafür, auf Assembler-Programmierung umzusteigen. Vorrangig dürfte aber die extreme Geschwindigkeitssteigerung gegenüber einer Hochsprache, wie beispielsweise Basic, sein. Auch diejenigen, die sich lieber mit der Hardware eines Computers beschäftigen, greifen vorzugsweise auf Assemblerprogrammierung zurück. Denn nicht selten ist es nur dann möglich, bestimmte Bauteile im Computer anzusprechen und zu programmieren. Wer beispielsweise den Antic, also den speziellen Grafikprozessor im 800XL/130XE, optimal nutzen möchte, kommt an Assembler nicht vorbei.

Übrigens sind Assembler und Maschinensprache nicht dasselbe. Wer nämlich in Maschinensprache programmiert, der ordnet jeder einzelnen Speicherzelle im RAM-Speicher einen bestimmten Hexadezimalcode zu. Es wird also nur mit Zahlenwerten gearbeitet. In Assembler andererseits programmiert man mit leichter verständlichen Befehlen, den sogenannten Mnemonics. Mit dem Editor eines Assemblers gibt man Befehle wie LDA oder CMP ein. Diese werden dann vom Assembler in Maschinencode umgewandelt. Erst wenn dieser Code im Speicher vorliegt, kann der Microprozessor die Befehle abarbeiten.

Aber nicht allein Zeitgewinn oder auch Platzersparnis sind gute Gründe für die Anwendung von Maschinensprache. Um beispielsweise die Fähigkeiten des Atari-Betriebssystems auszureizen, muß man zumindest einige Programmteile in Maschinensprache schreiben.

Ob man eine kleine Unterroutine für ein Basic-Programm oder ein ganzes Programm vollständig in Maschinensprache schreiben will, ein Assembler ist auf jeden Fall eine lohnende Investition.

Von der Firma OSS (Optimized Systems Software) gibt es für etwa 250 Mark das Programmodul »MAC/65«. Es enthält alle wesentlichen Bestandteile eines Assemblers – nämlich einen zeilennummernorientierten Editor, ähnlich wie in Basic, den eigentlichen Assembler und einen Monitor, über den man seine Maschinensprachprogramme austesten kann.

Die Modulform bietet mehrere wichtige Vorteile gegenüber einem Programm auf Diskette oder Kassette. So kann man beispielsweise nach Lust und Laune mit jeder Kombination von Diskettenlaufwerken arbeiten, da man ja in der Wahl des Diskettenbetriebssystems völlig frei ist. Außerdem muß man nicht nach einem eventuellen Systemabsturz, der ja bei Programmierung auf Maschinenspracheebene relativ schnell passieren kann, den Assembler neu laden. Darüber hinaus hat es OSS durch Umschaltung zwischen verschiedenen Speicherbänken geschafft, den Speicherplatzbedarf von MAC/65 auf 8 KByte zu beschränken.

Nun aber zum eigentlichen Programm – beginnen wir mit dem Editor:

Genau wie im normalen Atari-Basic hat man es mit einem zeilennummernorientierten Editor zu tun. Er ist jedoch um einige sehr wichtige Kommandos erweitert. So kann man mit FIND bestimmte Programmteile suchen, mit REP ersetzen, mit DEL Programmbereiche löschen und mit RENUM das gesamte Programm neu durchnumerieren. Auch ein Befehl zur automatischen Erzeugung von Zeilennummern fehlt nicht. Eine praktische Sache ist, daß MAC/65, genau wie Atari-Basic, die Programmzeilen unmittelbar nach Eingabe auf ihre Richtigkeit überprüft und sofort einen eventuellen Fehler anzeigt.

Nun zum Assembler, den man mit dem Kommando ASM startet. Dabei kann man noch angeben, woher – falls nicht aus dem RAM-Speicher – der Programmtext genommen werden soll, wohin die Ausgabe gehen soll und wohin der erzeugte Maschinencode geschrieben wird.

Selbstverständlich versteht MAC/65 alle normalen 6502-Mnemonics.

Datenfelder kann man einfach mit dem Befehl ».Byte« einfügen – wobei man entweder dezimale, hexadezimale oder ASCII-Darstellung wählen (und mischen) kann. Sehr nützlich ist dabei die Möglichkeit, zu allen errechneten Werten eine Konstante zu addieren. Weitergehende Befehle in dieser Richtung sind ».SBYTE« (Zeichen in interner Darstellung), ».WORD« (Doppelbyte), ».DBYTE« (Doppelbyte in umgekehrter Schreibweise), ».CBYTE« (Text mit invertiertem letzten Zeichen) und ».FLOAT« (Fließkommazahl). Der »MAC/65«-Assembler erlaubt auch das Arbeiten mit Markierungen (Labels) innerhalb eines Programms. Die Länge einer Zeichenkette läßt sich zum Beispiel folgendermaßen berechnen:

```
1000 TEXT .BYTE "Happy Computer"
1010 TEXT_LEN = *-TEXT
```

Auch zu Multiplikationen, Divisionen oder Bitoperationen in Ausdrücken ist der »MAC/65« fähig.

Nützlich ist die konditionelle Assemblierung. Mit Hilfe dieser Funktion lassen sich bei der Assemblierung bestimmte Programmteile von irgendeiner Bedingung abhängig machen. Hat man beispielsweise folgende Variablen festgelegt:

```
1000 DEUTSCH=1
1010 VERSION=DEUTSCH
```

dann kann man später mit

```
2000 .IF VERSION=DEUTSCH
2010 .BYTE "Bitte Diskette
     einlegen"
2020 .ELSE
2030 .BYTE "Please insert disk"
2040 .ENDIF
```

zwischen den verschiedenen Fällen unterscheiden. So braucht man keine verschiedenen Quelltexte, um unterschiedliche Versionen eines Programms zu schreiben.

In diesem Zusammenhang ist die Funktion ».REF« wichtig. Mit ihr kann man innerhalb einer ».IF«-Anweisung feststellen, ob ein bestimmtes Label innerhalb des Textes bereits benutzt wurde. Dies gestattet Unterroutinen nur dann mit einzubinden, wenn sie tatsächlich gebraucht werden. Zusammen mit der ».INCLUDE«-Funktion, mit der sich Programmteile von Diskette einbinden lassen, kann man auch Unterprogramme aus Bibliotheken nutzen.

Ein wichtiges Merkmal von »MAC/65« ist die Fähigkeit, Makros zu verarbeiten (daher auch der Name MAC/65). Mit Hilfe von Makros kann man mehrere normale Assemblerbefehle unter einem Wort zusammenfassen. Es lassen sich auch Parameter übergeben und verarbeiten. Ein Beispiel für eine einfache Makrodefinition:

```
1000 .MACRO JEQ
1010 BNE ?JEQ
1020 JMP %1
1030 ?JEQ
1040 .ENDM
```

Zunächst wird hierbei der Name des Makros festgelegt. »JEQ« soll hier einen Befehl simulieren, der im normalen 6502-Befehlssatz nicht vorhanden ist, nämlich einen konditionellen Sprungbefehl zu einer absoluten Adresse (in diesem Beispiel, falls die Abfrage zutrifft). Falls nicht (Branch Not Equal), wird an das Ende der Makrodefinition verzweigt. Das Programm läuft also beim nächstfolgenden Befehl weiter. Im anderen Fall wird ein Sprung an die als erster Parameter angegebene Adresse durchgeführt. Der Aufruf könnte dann so aussehen:

```
2000 LDA ZAEHLER
2010 CMP #HOECHSTWERT
2020 JEQ KEINE_VERAENDERUNG
```

Hier kann man übrigens auch sehr gut sehen, daß Assemblerprogramme nicht unbedingt schwer lesbar sein müssen. Immerhin kann man, soweit es der Speicherplatz erlaubt, die Labelnamen beliebig lang und damit ziemlich selbsterklärend gestalten.

Ein weiteres Beispiel wäre ein Makro zur Addition von 16-Bit-Werten, für die es bekanntlich beim 6502 keinen eigenen Befehl gibt:

```
1000 .MACRO AIW ;Add Immediate
     to Word
1010 .IF %2=2   ;Eins addieren?
1020 INC %1
1030 BNE ?NO_INC
1040 INC %1+1
1050 NO_INC
1060 .ELSE
1070 LDA %1
1080 CLC
1090 ADC #< %2
1100 STA %1
1110 LDA %1+1
1120 ADC #> %2
1130 STA %1+1
1140 .ENDIF
1150 .ENDM
```

Hier wird zuerst geprüft, ob das zweite Argument, also die Zahl, die addiert werden soll, den Wert 1 enthält. In diesem Fall kann der etwas kürzere Weg über die Inkrementierung der beiden Bytes gegangen werden. In allen anderen Fällen (».ELSE«-Abfrage) werden der Reihe nach zunächst das niederwertigere und dann das höherwertigere Byte addiert. Man beachte, daß Makros auch eine Optimierung des Programmtextes vornehmen können, wenn sie sinnvoll konditionell definiert sind. Obwohl im Programmcode nun jedes Mittel zur Kürzung des Codes vorgenommen werden kann, sieht der tatsächliche Quelltext wesentlich übersichtlicher aus – und ist zudem noch kürzer! Was will man mehr? Auch hier ein Beispiel für die Benutzung dieses Makros:

```
2000 AIW PNT,7680
```

Man vergleiche dies mit:

```
2000 LDA PNT
2010 CLC
2020 ADC #< 7680
2030 STA PNT
2040 LDA PNT+1
2050 ADC #> 7680
2060 STA PNT+1
```

Mit Makros kann man aber nicht nur oft benutzte Kombinationen von Befehlen, sondern auch Befehlsfolgen, die in einem eigenen Programm regelmäßig auftreten, verkürzen. Auch zum Aufbau von Datentabellen sind Makros nützlich, da sich auch die übergebenen Parameter direkt in Daten umwandeln lassen.

Ein weiterer Pluspunkt von Makros ist, daß es Umsteigern von Basic erleichtert wird, in Assembler zu programmieren – Makrobibliotheken, die Makros wie »PRINT«, »INPUT« oder »GRAPHICS« enthalten, gibt es zur Genüge.

Kommen wir zum eigentlichen Assemblerteil von »MAC/65«. Hier lassen sich für den Quelltext, das Listing und den erzeugten Code beliebige Peripheriegeräte festsetzen. Dabei kann man den Quelltext auch direkt aus dem Speicher übernehmen. Oder man legt den generierten Maschinencode direkt im RAM-Speicher ab. Dabei muß man natürlich darauf achten, den im Speicher befindlichen Programmtext nicht zu überschreiben. Das Wichtigste ist natürlich die Assembliergeschwindigkeit. Dazu ein Beispiel: Für etwa 100 KByte Quelltext (wobei zu beachten ist, daß MAC/65 die Befehle in Tokens verwandelt und daher relativ wenig Platz für Quelltexte verbraucht) benötigt der Assembler etwa drei Minuten. In diesem Zeitraum wurde der gesamte Quelltext von Diskette gelesen und der erzeugte Code wiederum auf Diskette geschrieben. Hier darf man nicht verschweigen, daß die meiste Zeit beim Assemblieren durch Zugriffe auf externe Speichermedien verlorengeht. Deshalb lassen sich bessere Ergebnisse erzielen, wenn man auf dem Atari 130 XE mit der RAM-Disk unter DOS 2.5 arbeitet.

Ein anderer wichtiger Gesichtspunkt ist, daß der »MAC/65«-Assembler, nachdem er mehrmals verbessert wurde, als fast fehlerfrei bezeichnet werden kann. Dies bestätigt mehrjährige Erfahrung. Was wäre schlimmer als ein Assembler, der bereits beim Assemblieren Fehler in den Maschinencode bringt?

Doch der Assemblerteil von »MAC/65« hat noch einen weiteren Leckerbissen zu bieten: Für die Ausgabe des Protokolls beim Assemblieren auf Druckern stehen vielfältige Funktionen zur Verfügung. Man kann sogar den Inhalt und die Form des Listings beeinflussen. So ist es zum Beispiel kein Problem, die Erzeugung des Listings für bestimmte Teile des Programms abzuschalten oder bei Makros alle nicht wichtigen Teile wegzulassen. Außerdem lassen sich die Seitenlänge und -breite sowie der Steuercode für Seitenvorschub festlegen. Diese Angaben befähigen den »MAC/65«, ein übersichtliches Listing mit Seitennumerierung und Seitenüberschriften zu erzeugen. Wahlweise kann auch noch eine Cross-Reference-Liste ausgestellt werden. So könnte in alphabetischer Reihenfolge für jedes benutzte Label nicht nur der erzeugte Wert, sondern auch jedes Auftauchen im Programm mit Seiten- und Zeilennummer angegeben werden. Eine sehr nützliche Funktion, um zeitaufwendige Fehlersuche zu ersparen.

Der dritte Bestandteil vom »MAC/65«-Assembler ist der Monitor. Er nennt sich »DDT« (Dunions Debugging Tool), da er durch Eingabe von DDT aufzurufen ist. Hier sieht man sofort den ersten großen Unterschied zu konventionellen Debuggern. Statt einen normalen Textbildschirm zu verwenden, erzeugt er ein spezielles Bildschirmformat. Dieses ist sehr übersichtlich und informativ und beinträchtigt trotzdem die Übersichtlichkeit des zu bearbeitenden Programmbildes nicht. Auf dem verhältnismäßig kleinen Bildschirmausschnitt kann ein Teil des Speichers in disassemblierter Form oder in hexadezimaler Schreibweise betrachtet werden. Weiterhin erfolgt noch die Ausgabe der Positionen der vier Breakpoints sowie der Prozessor- und Statusregister. Hier fehlen zwar Ein- und Ausgabefunktionen, dafür gibt es aber einige unübliche Bonbons. So kann man mit START das Programm an der Stelle fortsetzen, an der sich der PC (Program-Counter) befindet. Mit SELECT erfolgt ein Umschalten zwischen dem DDT-Bildschirm und dem vom Programm erzeugten Bildschirm, und mit OPTION lassen sich noch einzelne Maschinensprachbefehle abarbeiten.

Eine lästige Eigenschaft von DDT soll hier nicht verschwiegen werden. Da der Speicherausschnitt im Bildschirmfenster nur mit den Cursortasten um einzelne Zeilen verschoben werden kann, ist es leider ein Unding, bestimmte Programmteile schnell zu finden.

Zusammenfassend kann man jedoch sagen: *DDT ist zum Bearbeiten von Programmen, die sich bereits im Speicher befinden, prädestiniert.*

Als Zusatz zum »MAC/65«-Assembler wird von OSS eine Diskette mit Hilfsprogrammen und vordefinierten Makros (»MAC/65 Toolkit«) angeboten. Hier findet man beispielsweise Makros für alle normalen Ein- und Ausgabefunktionen, normale und Player/Missile-Grafik und Arithmetik. Darunter befindet sich auch ein Programm, das die Kommandozeile von DOS XL analysiert und somit das Schreiben von zusätzlichen Kommandos für DOS XL wesentlich erleichtert. Ein komplettes Dateikopierprogramm, das fast ausschließlich mit Makros geschrieben ist, fehlt auch nicht.

Man kann mit gutem Gewissen sagen, daß MAC/65 für jeden, der sich in Assembler versuchen will, hervorragend geeignet ist. Dabei kommen auch Anfänger nicht zu kurz, die sich in der Basic-ähnlichen Umgebung wohlfühlen werden. Ein fortgeschrittener Programmierer wird nach einigen Stunden Arbeit mit »MAC/65« nicht mehr auf diesen komfortablen Assembler verzichten wollen. (Julian F. Reschke/wb)

Bezugsquellen:
Münzenloher, Tölzer Str. 4, 8150 Holzkirchen, Tel. (08024) 1814
Compy-Shop, Gneisenaustr. 29, 4330 Mülheim, Tel. (0208) 497169

»SynFile+«, Dateiverwaltung total

Vor allem durch eine gelungene Benutzerführung und ein gut durchdachtes Konzept hebt sich »SynFile+« angenehm von anderen Dateiverwaltungen ab.

Hat man schon mal einen Heimcomputer zu Hause stehen, sollte man ihn nicht nur zum Spielen nutzen. Schließlich gibt es eine Reihe von sinnvollen Anwendungen, die sich auch auf einem Atari realisieren lassen. So zum Beispiel eine Dateiverwaltung wie »SynFile+«. Mit ihr kann man Daten aller Art übersichtlich und sauber verwalten. Sei dies nun ein Adressenverzeichnis, eine Plattensammlung oder Mutters Kochrezepte, das spielt keine Rolle. Aber auch Kundendaten oder Verkaufsstatistiken kann man mit ein und demselben Programm verwalten.

»SynFile+« wird auf Diskette geliefert und ist für Atari Computer mit mindestens 48 KByte Speicher gedacht. Man erhält noch eine rund 120 Seiten umfassende Bedienungsanleitung in englischer Sprache, die jedoch durch Aufmachung und Gliederung gut zu handhaben ist. Auch die Benutzerführung des Programms ist natürlich in Englisch gehalten. Dies stört aber nicht so sehr, da die Bedienung von »SynFile+« vollständig menügesteuert ist.

Das Funktionsprinzip von »SynFile+« ist eigentlich ganz einfach. Am besten stellt man sich das Programm als Karteikasten vor. Jede Karteikarte, auf der man Daten eintragen kann, umfaßt 21 Zeilen mit je 80 Spalten. Auf dieser Karte trägt man zuerst die benötigten Datenfelder ein. Welche Felder man verwendet und wie man sie benennt, hängt natürlich vom eigenen Bedarf ab. Baut man sich beispielsweise ein Adressenverzeichnis auf, so würden die Felder »Name«, »Vorname«, »Wohnort«, »Telefon« und so weiter lauten. Wie man diese Felder auf der Karte plaziert, bleibt dabei den eigenen Wünschen und Vorstellungen überlassen. Bei jedem Feld gibt man zudem noch die maximale Länge an, die jeder Eintrag umfassen darf. Außerdem muß der Typ des jeweiligen Feldes festgelegt werden. So soll ein Name beispielsweise später nur aus Buchstaben bestehen. Andererseits setzt sich eine Telefonnummer aus numerischen Werten zusammen. Die Typenzuweisung birgt gleich zwei Vorteile in sich. Einmal kann das Programm dadurch leichter auf spätere Falscheingaben reagieren. Gibt man beispielsweise bei einer Telefonnummer versehentlich einen Buchstaben ein, so wird diese Eingabe aufgrund des nicht übereinstimmenden Feldtyps nicht akzeptiert. Zudem verfügt man bei »SynFile+« über eine Reihe von Feldtypen, die das Arbeiten mit dem Programm wesentlich komfortabler machen. So gibt es beispielsweise den Typ »Date«, also Datum. Hierbei sind nur Eingaben erlaubt, die tatsächlich auch einen Bezug zum Datum haben. Gibt man also »99/11/85 ein, reagiert »SynFile+« daraufhin mit einer Fehlermeldung. Allerdings muß ein Datum bei »SynFile+« in der amerikanischen Notierung, also erst der Monat, dann der Tag und zuletzt noch das Jahr, eingegeben werden.

Der Typ »Look-Up« wiederum erlaubt nur die Eingabe solcher Werte und Texte, die man vorher in einer Tabelle exakt festgelegt hat. Soll in einem Feld

Bild 1. So kann beispielsweise eine selbst aufgebaute Eingabemaske mit »SynFile+« aussehen

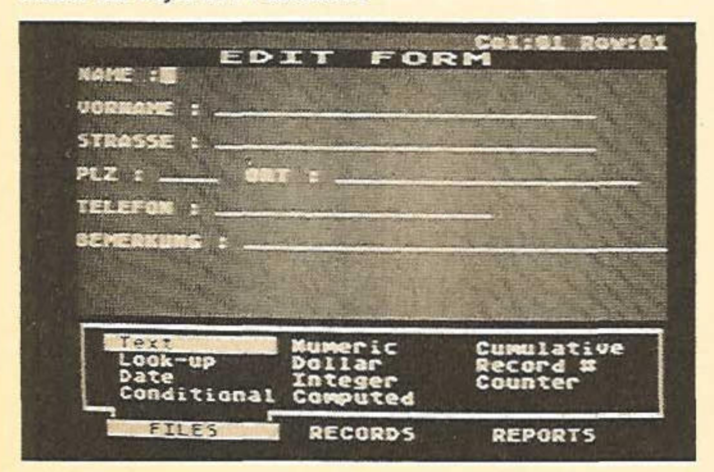
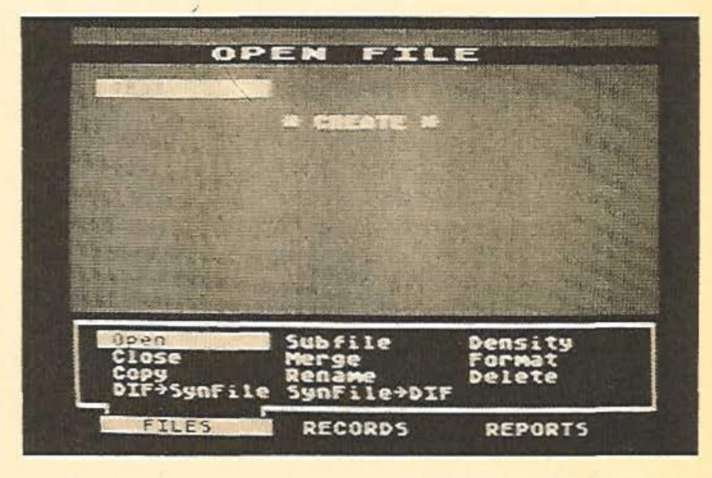


Bild 4. In allen Menüs von »SynFile+« wählt man die Unterpunkte mit Hilfe der Cursortasten



```
NAME: Wurm
VORNAME: Peter

STRASSE: Peterstr.          NR.:  11

PLZ: 2222  ORT: Wurmhausen

TELEFON: 2345

GEBURTSTAG: 01/01/01 GESCHLECHT: M

BEMERKUNG: wurmt
```

Bild 2. Ein Datensatz als Hardcopy

GESAMTLISTE

NAME	VORNAME	STRASSE	NR.	PLZ	ORT
Czerny	Wolfgang	Aschenstrasse	100	2222	Muenchen 12
Kohlrabi	Rotkohl	Weisskraut	1	4711	Sauerkraut
Markt	und	Technik	2	8013	Haar
Mueller	Hans	Dorfstrasse	10	1111	Gongdorf
Schraube	Nagel	Duebelstr.	2	2222	Holwurmhausen
Wurm	Peter	Petersstr.	11	2222	Wurmhausen

Bild 3. Eine Liste aller Datensätze läßt sich auf einem Drucker ausgeben

zum Beispiel das Geschlecht einer Person eingetragen werden, so legt man in der Tabelle die Eingaben »m« für männlich und »w« für weiblich fest. Alle anderen Eingaben werden dann ignoriert. Mit Hilfe des Typs »Conditional« kann man dann noch einen Feldinhalt von einer vorher definierten Bedingung abhängig machen. Soll mit dem Inhalt eines oder mehrerer Felder gerechnet werden, kommt das Ergebnis dieser Berechnungen in ein Feld des Typs »Computed«.

Sobald die Definition aller Felder abgeschlossen ist, hat man eine fertige Eingabemaske, wie sie in Bild 1 zu sehen ist. Sollte man feststellen, daß die Maske noch nicht in der gewünschten Form vorliegt, kann man sie natürlich auch nachträglich noch ändern. Man ordnet der Datei einfach einen anderen Dateinamen zu und speichert sie auf Diskette. Als nächsten Schritt indiziert man ein oder mehrere Felder eines Datensatzes. Bei der Indizierung wird der Inhalt des jeweiligen Feldes in ein eigenes File abgelegt. Auf diese Weise muß das Programm beim Suchen oder Sortieren von Datensätzen nicht die gesamte Datei, sondern nur die jeweilige Indexdatei durchforsten. Bis zu 16 Felder können gleichzeitig indiziert werden. Dabei ist jedoch darauf zu achten, daß der Speicherplatz auf Diskette mit der Zahl der Indexfelder rapide abnimmt. Auch ist die Zahl der Datensätze, die eine Datei umfassen kann, vorwiegend durch die Größe des Indexfeldes begrenzt. Im Gegensatz zu den eigentlichen Daten muß das Indexfeld nämlich im Speicher des Computers Platz finden. Die Datensätze einer einzigen Datei hingegen können auf bis zu 16 Disketten abgelegt werden.

Hat man seine Daten eingegeben, kann man damit natürlich einige Manipulationen vornehmen. So lassen sich die Datensätze nach steigender oder fallender Folge sortieren. Einzelne Datensätze kann man auch mit Hilfe von verschiedenen Suchkriterien schnell finden. Gibt man mehrere Suchkriterien an, kann man wählen, ob alle Bedingungen gleichzeitig oder nur eines der Kriterien erfüllt sein muß.

Will man mit Hilfe einer Textverarbeitung Serienbriefe drucken, können die Daten, die in die Briefe eingefügt werden sollen, natürlich einer SynFile-Datei entnommen werden. Ein Beispiel: Man schickt ein Rundschreiben an alle Personen, die in einer Adreßdatei gespeichert sind und in einer bestimmten Stadt wohnen. Anschließend gibt man noch als Suchkriterium die gewünschte Stadt ein. Aus den gefundenen Datensätzen werden nun Namen und Adressen in ein separates File geschrieben. Eine Textverarbeitung wie der »Atari-Schreiber« beispielsweise, kann nun diese Datei lesen und deren Daten in die einzelnen Briefe eintragen.

Die Inhalte einer Datei kann man natürlich auch auf einem Drucker ausgeben. Dies kann grundsätzlich auf drei Arten geschehen. Einmal kann man quasi eine Hardcopy des momentan auf dem Bildschirm befindlichen Datensatzes anfertigen (Bild 2). Will man mehrere Datensätze ausdrucken, kann man die einzelnen Felder entweder nebeneinander oder untereinander darstellen (Bild 3). Auch beim Ausdruck kann man sich die jeweiligen Datensätze mit Hilfe von Suchkriterien beliebig zusammenstellen. Auf diese Weise lassen sich auch ohne weiteres Adreßetiketten beschriften. Leider sieht »SynFile+« nicht die Möglichkeit vor, eine Druckmaske zu speichern.

Die Benutzerführung von »SynFile+« ist als vorbildlich zu bezeichnen. Am unteren Bildrand findet man drei Hauptmenüpunkte. Mit Hilfe der Cursortasten wählt man sich den jeweiligen Punkt aus. Drückt man dann RETURN, so erscheint das jeweilige Untermenü in Form eines Fensters auf dem Bildschirm (Bild 4). Die einzelnen Untermenüpunkte werden wiederum mit den Cursortasten angewählt. Auch die Typenzuweisung bei der Felderdefinition wird über ein Untermenü durchgeführt. Mit diesem Bedienungsschema erreicht »SynFile+« eine Übersichtlichkeit, wie sie bei kaum einem Programm für Atari-Computer zu finden ist.

Mit Ausnahme der etwas mangelhaften Druckeranpassung kann man »SynFile+« also kaum Schwächen nachsagen. Ein echtes Kaufhindernis dürfte jedoch der Preis dieses Programms sein. Bei rund 220 Mark sollte man doch eine rentable Anwendung für dieses Programm vorweisen können.

(Wolfgang Czerny/wb)

Bezugsquellen:
CompyShop, Gneisenaustr. 29, 4330 Mühlheim/Ruhr, Tel. (0208) 497169
Münzenloher, Tölzer Str. 4, 8150 Holzkirchen, Tel. (08024) 1814









J. Schultz/W. Pest

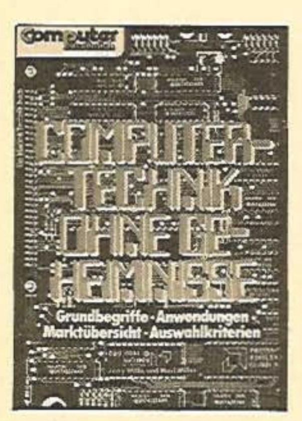
Drucker-Handbuch

Januar 1985, 188 Seiten

Welchen Drucker brauche ich? Einen schnellen, einen leisen, einen billigen, einen korrespondenzfähigen? Welcher Drucker kann problemlos an meinen Personal- oder Homecomputer angeschlossen werden? Wie funktioniert so ein Drucker überhaupt?
Lesen Sie dieses Buch, und Sie wissen auf alle diese Fragen eine fundierte Antwort!

• Endlich ein informativer Leitfaden für alle, die vor dem Kauf eines Druckers stehen.

Best.-Nr. MT 742
ISBN 3-89090-077-1 **DM 38,-**



J. Willis/M. Miller

Computertechnik ohne Geheimnisse

November 1984, 313 Seiten

Ob Sie schon einen Computer besitzen, seine Anschaffung planen oder sich nur mal aus Interesse mit diesem faszinierenden Gebiet auseinandersetzen wollen: in diesem Buch finden Sie die Antwort auf alle Fragen zur Computertechnik. Das Buch beschreibt den derzeitigen Stand der Technik, gibt Aufschluß über die Entwicklung der Computer und verschafft Ihnen eine Marktübersicht über Hard- und Software (wenn Sie nicht wissen, was das ist, dann lesen Sie dieses Buch!). Und es beantwortet auf amüsante Weise die oft gestellte Frage: »Was, zum Kuckuck, kann ich eigentlich mit so einem Ding machen?«.

• Informationen zum Thema Computer von allgemeinem Interesse.

Best.-Nr. MT 716
ISBN 3-89090-066-6 **DM 42,-**

Markt & Technik-Fachbücher erhalten Sie bei Ihrem Buchhändler.

Markt&Technik
BUCHVERLAG

Hans-Pinsel-Straße 2, 8013 Haar bei München

Depot-Händler

Tragen Sie Ihre Buchbestellung auf eine Postkarte ein und schicken diese an einen Depothändler in Ihrer Nähe oder an Ihren Buchhändler.

Buchhandlung Herder, Kurfürstendamm 69
1000 Berlin 15, Tel. (030) 8835002,
BTX *921782#
Computare Fachbuchhandlung, Keithstraße 18
1000 Berlin 30, Tel. (030) 2139021
Thalia Buchhaus, Große Bleichen 19
2000 Hamburg 36, Tel. (040) 3005050
Boysen + Maasch, Hermannstraße 31
2000 Hamburg 1, Tel. (040) 30050515
Electro-Data, Wilhelm-Heidsiek-Straße 1
2190 Cuxhaven, Tel. (04721) 51288
Buchhandlung Muehlau, Holtenauer Straße 116
2300 Kiel, Tel. (0431) 85085
ECL, Norderstraße 94-96
2390 Flensburg, Tel. (0461) 28181
Buchhandlung Weiland, Königstraße 79
2400 Lübeck, Tel. (0451) 74006-09
Buchhandlung Storm, Langenstraße 10
2800 Bremen 1, Tel. (0421) 321523
Buchhandlung Lohse-Eissing, Marktstraße 38
2940 Wilhelmshaven, Tel. (04421) 41687
Buchhandlung Schmorl u. v. Seefeld, Bahnhofstraße 13
3000 Hannover 1, Tel. (0511) 327651
Buchhandlung Graff, Neue Straße 23
3300 Braunschweig, Tel. (0531) 49271
Deuerlich'sche Buchhandlung, Weender Straße 33
3400 Göttingen, Tel. (0551) 56868
Buchhandlung an der Hochschule, Holländische Straße 22
3500 Kassel, Tel. (0561) 83807
Stern Verlag, Friedrichstraße 24-26
4000 Düsseldorf, Tel. (0211) 373033
Buchhandlung Baedeker, Kettwiger Straße 33-35
4300 Essen 1, Tel. (0201) 221381
Regensberg'sche Buchhandlung, Alter Steinweg 1
4400 Münster, Tel. (0251) 40541-5
Buchhandlung Acker, Johannisstraße 51
4500 Osnabrück, Tel. (0541) 28488
Buchhandlung Lensing, Westenhellweg 86-88
4600 Dortmund, Tel. (0231) 16980
Buchhandlung Brockmeyer, Querenburger Höhe 281/Unicenter
4630 Bochum, Tel. (0234) 701360
Buchhandlung Meier + Weber, Warburger Straße 98
4790 Paderborn, Tel. (05251) 63172
Buchhandlung Phönix GmbH, Oberntorwall 25
4800 Bielefeld 1, Tel. (0521) 69071
Buchhandlung Gonski, Neumarkt 24
5000 Köln 1, Tel. (0221) 210528
Meyer'sche Buchhandlung, Ursulinerstraße 17-19
5100 Aachen, Tel. (0241) 48142
Buchhandlung Behrendt, Am Hof 5a
5300 Bonn 1, Tel. (0228) 658021
Buchhandlung Cusanus, Schloßstraße 12
5400 Koblenz, Tel. (0261) 36239
Akad. Buchhandlung Interbook, Fleischstraße 61-65
5500 Trier, Tel. (0651) 43596
Buchhandlung W. Finke, Kipdorf 32
5600 Wuppertal 1, Tel. (0202) 454220
Buchhandlung Balogh, Sandstraße 1
5900 Siegen, Tel. (0271) 55298-9
Buchhandlung Naacher, Steinweg 3
6000 Frankfurt 1, Tel. (069) 298050
Buchhandlung Wellnitz, Lautenschlägerstraße 4
6100 Darmstadt, Tel. (06151) 76548
Buchhandlung Feller + Gecks, Friedrichstraße 31
6200 Wiesbaden, Tel. (06121) 304911
Ferber'sche UNI-Buchhandlung, Seltersweg 83
6300 Gießen, Tel. (0641) 12001
Sozialwissenschaftliche Fachbuchhandlung, Friedrichstraße 24
6400 Fulda, Tel. (0661) 75077
Gutenberg Buchhandlung, Große Bleiche 29
6500 Mainz, Tel. (06131) 37011
Buchhandlung Bock + Seip, Futterstraße 2
6600 Saarbrücken, Tel. (0681) 30677
Buchhandlung Wilhelm Hofmann, Bismarckstraße 98
6700 Ludwigshafen, Tel. (0621) 516001
Buchhandlung Loeffler, B 1,5
6800 Mannheim 1, Tel. (0621) 28912
Buchhandlung Stehn, Bahnhofstraße 13
7000 Stuttgart 50, Tel. (0711) 561476
Buchhandlung am Markt, Kramstraße 6
7100 Heilbronn, Tel. (07131) 68682
PCB Micro-Computer, Oskar-Kalbfell-Platz 8
7410 Reutlingen, Tel. (07121) 270443
UNI Buchhandlung Kellner + Moessner, Kaiserstraße 18
7500 Karlsruhe, Tel. (0721) 691436
Buchhandlung Roth, Hauptstraße 45
7600 Offenburg, Tel. (0781) 22097
Rombach Center, Bertholdstraße 10
7800 Freiburg, Tel. (0761) 49091
Fachbuchhandlung Hofmann, Hirschstraße 4
7900 Ulm, Tel. (0731) 60949
Schauties Elektronik, Bachstraße 52
7980 Ravensburg, Tel. (0751) 26138
Buchhandlung Hugendubel, Marienplatz
8000 München 2, Tel. (089) 2389-1
Computerbücher am Obelisk, Barerstraße 32-34
8000 München 2, Tel. (089) 282383
Pele's Computerbücher, Schillerstraße 17
8000 München 2, Tel. (089) 555229
Universitätsbuchhandlung Lachner, Theresienstraße 43
8000 München 2, Tel. (089) 521340
Buchhandlung Schönhuber, Theresienstraße 6
8070 Ingolstadt, Tel. (0841) 33146/47
Computerstudio Gertrud Friedrich, Ludwigstraße 3
8220 Traunstein, Tel. (0861) 14767
Buchhandlung Pustet, Kl. Exerzierplatz 4
8390 Passau, Tel. (0851) 56945
Buchhandlung Pustet, Gesandtenstraße 6
8400 Regensburg, Tel. (0941)53061
Buchhandlung Dr. Büttner, Adlerstraße 10-12
8500 Nürnberg, Tel. (0911) 232318
STS Computer Vertrieb, Werner-Siemens-Straße 19
8580 Bayreuth, Tel. (0921) 62320
Computer-Center-Burger, Leimitzer Straße 11-13
8670 Hof, Tel. (09281) 40075
Sortiments- u. Bahnhofsbuchh. J. Strykowski, Bahnhofplatz 4
8700 Würzburg, Tel. (0931) 54389
Buchhandlung Pustet, Grottenau 4
8900 Augsburg, Tel. (0821) 35437
Kemptener Fachsortiment, Salzstraße 30
8960 Kempten, Tel. (0831) 14413

Belgien:
Eicher Micro & Personal Computer, Hünningen 56-58
B-4780 St. Vith, Tel. (080) 227393

Luxemburg:
Librairie Promoculture, 14, rue Duchscher (Pl. de Paris)
L-1011 Luxembourg-Gare, Tel. 480691, Telex 3112

Schweiz:
Buchhandlung Meissner, Bahnhofstraße 41
5000 Aarau, Tel. (064) 247151
Bücher Balmer, Neugasse 12
6300 Zug, Tel. (042) 214141
Buchhandlung Enge, Bleicherweg 56
8002 Zürich, Tel. (01) 2012078
Buchhandlung Orell Füssli, Pelikanstraße 10
8022 Zürich, Tel. (01) 2118011
Freihofer AG, Wissenschaftliche Buchhandlung, Universitätsstr. 11
8033 Zürich, Tel. (01) 3634282
Buchhandlung am Rößlitor, Webergasse 5
9001 St. Gallen, Tel. (071) 228726

Markt&Technik
BUCHVERLAG

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael Scharfenberger (sc)
Leitender Redakteur: Michael Lang (lg)
Redakteure: Werner Breuer (wb; Inhalt)
Petra Wängler, Eva Hierlmeier (Koordination)
Redaktionsassistenz: Monika Lewandowski (222)

Fotografie: Jens Jancke

Layout: Leo Eder (Ltg.),
Sigrid Kowalewski (Cheflayouterin)

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Tel. (042) 415656, Telex: 862329 mut ch
USA: M&T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. 415-424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Produktionsleitung: Klaus Buck (180)

Anzeigenverkauf: Brigitta Fiebig (211)

Anzeigenverwaltung und Disposition:
Patricia Schiede (172)

Marketingleiter Vertrieb: Hans Hörl (114)

Vertriebsleitung: Helmut Grünfeldt (189)

Verlagsleiter M&T Buchverlag: Günther Frank (212)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Str. 96, 7000 Stuttgart 1, Tel. (0711) 6483-0

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Bezugspreis: Das Einzelheft kostet DM 14,-.

Druck: Druckhaus München GmbH, Schellingstraße 39-43, 8000 München 40

Verantwortlich: Für redaktionellen Teil:
Michael Scharfenberger
Für Anzeigen: Ralph Peter Rauchfuß (126).

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft,
Hans-Pinsel-Straße 2, 8013 Haar bei München,
Telefon (089) 4613-0, Telex 5-22052

Telefon-Durchwahl im Verlag:
Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Aktionäre, die mehr als 25% des Kapitals halten: Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München; Aufsichtsrat: Dr. Robert Dissmann (Vorsitzender), Karl-Heinz Fanselow, Eduard Heilmayr



Markt&Technik
ATARI ST-Software



mit MailMerge für die
ATARI ST-Computer

3½"-Format

WordStar für den ATARI ST

Der Bestseller unter den Textverarbeitungsprogrammen bietet Ihnen bildschirmorientierte Formatierung, deutschen Zeichensatz und DIN-Tastatur sowie integrierte Hilfstexte. Mit MailMerge können Sie Serienbriefe mit persönlicher Anrede an eine beliebige Anzahl von Adressen schreiben und auch die Adreßaufkleber drucken.

Jetzt gibt es WordStar/MailMerge für den ATARI ST! Damit eröffnen sich Ihnen alle Möglichkeiten, Ihren ATARI ST für professionelle Textverarbeitung einzusetzen. Zum Superpreis!

WordStar für den ATARI ST wird auf einer 3½-Zoll-Diskette geliefert.
Sie beinhaltet:

- CP/M-Z 80-Emulator
- WordStar/MailMerge-Dateien

Hardware-Anforderungen: ATARI ST-Computer, 80-Zeichen-Monitor, ein 3½"-Diskettenlaufwerk, beliebiger Drucker mit Centronics-Schnittstelle.

WordStar ist an den ATARI ST bereits fertig angepaßt und läßt sich bequem über Funktionstasten steuern.

Bestell-Nr. MS 106
Für sagenhafte DM 199,–* (sFr. 178,–)

*inkl. MwSt. Unverbindliche Preisempfehlung.

Markt&Technik-Softwareprodukte erhalten Sie in den Computer-Abteilungen der Kaufhäuser und im Computershop.

Wenn Sie direkt beim Verlag bestellen wollen: Nur per Nachnahme, gegen Vorauskasse, Verrechnungsscheck oder mit der eingehefteten Zahlkarte in diesem Heft.

Bestellungen im Ausland: Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, ☎ 042/415656; Österreich: Ueberreuter Media Handels- und Verlagsges. mbH, Alser Straße 24, 1091 Wien, Tel. 0222/481538-0

Markt&Technik
Atari ST-Software
Hans-Pinsel-Straße 2, 8013 Haar bei München